(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A COMPARISON OF THE K-PULSE AND E-PULSE TECHNIQUES FOR ASPECT INDEPENDENT RADAR TARGET IDENTIFICATION

by

Martin S. Simon

September 1989

Thesis Advisor: M. A. Morgan

Approved for public release; distribution is unlimited.

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification UNCLASSIFIED	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report Approved for public release; distribution is unlimited
2b Declassification/Downgrading Schedule	
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School / 6b Office Symbol *(If Applicable)* 3A	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization / 8b Office Symbol *(If Applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*	10 Source of Funding Numbers: Program Element Number / Project No / Task No / Work Unit Accession No

11 Title *(Include Security Classification)* A COMPARISON OF THE K-PULSE AND E-PULSE TECHNIQUES FOR ASPECT INDEPENDENT RADAR TARGET IDENTIFICATION

12 Personal Author(s) Martin S. Simon

13a Type of Report	13b Time Covered	14 Date of Report *(year, month,day)*	15 Page Count
Master's Thesis	From To	1989, September	

16 Supplementary Notation The views expressed in this paper are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	E-Pulse; K-Pulse; Radar Target Identification; Natural Resonance; Resonance Annihilation; Noncooperative Target Recognition

19 Abstract *(continue on reverse if necessary and identify by block number*

It has been demonstrated that under laboratory conditions the natural resonant frequencies of a metallic scattering body may be used as the basis for target identification techniques. This thesis continues research into two such techniques: the difference equation based K-Pulse and the integral equation based E-Pulse. A double Gaussian smoothing function has been used in conjunction with the K-Pulse and basis functions of varying widths have been used in formulating the E-Pulse to provide enhanced performance in low signal to noise level environments. Digitally noise polluted synthetic scattered signals are used to analyze the effectiveness of the two techniques in distinguishing high, medium and low-Q targets. Both methods are shown to merit further study as potential candidates for implementation as Resonance Annihilation Filters for real-time radar target identification in a real-world environment. Theses,(

20 Distribution/Availability of Abstract [X] unclassified/unlimited [ ] same as report [ ] DTIC users	21 Abstract Security Classification Unclassified
22a Name of Responsible Individual M. A. Morgan	22b Telephone *(Include Area code)* (408) 646-2677 / 22c Office Symbol 62Mw

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted security classification of this page

All other editions are obsolete Unclassified

A Comparison of the K-Pulse and E-Pulse Techniques for Aspect Independent Radar Target Identification

by

Martin S. Simon
Lieutenant, United States Navy
A. B., University of Chicago, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING
(ELECTRONIC WARFARE)

from the

NAVAL POSTGRADUATE SCHOOL
September 1989

Author: Martin S. Simon
Martin S. Simon

Approved by: M. A. Morgan
M. A. Morgan, Thesis Advisor

James R. Powell
Commander James R. Powell, USN, Second Reader

Joseph Sternberg
Joseph Sternberg, Chairman,
Electronic Warfare Academic Group

ABSTRACT

It has been demonstrated that under laboratory conditions the natural resonant frequencies of a metallic scattering body may be used as the basis for target identification techniques. This thesis continues research into two such techniques: the difference equation based K-Pulse and the integral equation based E-Pulse. A double Gaussian smoothing function has been used in conjunction with the K-Pulse and basis functions of varying widths have been used in formulating the E-Pulse to provide enhanced performance in low signal to noise level environments. Digitally noise polluted synthetic scattered signals are used to analyze the effectiveness of the two techniques in distinguishing high, medium and low-Q targets. Both methods are shown to merit further study as potential candidates for implementation as Resonance Annihilation Filters for real-time radar target identification in a real-world environment..

TABLE OF CONTENTS

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verificatior is at the risk of the user.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

In modern naval warfare, positive identification of all potential targets within the zone of combat is essential to the protection of friendly forces. Since World War II, the United States Navy has been required to separate friendly and enemy aircraft in a complex radar environment, typically through the use of pre-planned return to force profiles and Identification Friend or Foe (IFF) systems. Unfortunately, these procedures have failed to completely preclude the possibility of "blue-on-blue" engagements, where friendly forces have brought their own aircraft under fire.

In peacetime scenarios, where visual identification and confirmation of enemy targets is often required, friendly forces could be at a significant disadvantage if the enemy chooses to launch an attack from beyond visual range. In Third World situations, early attainment of positive identification may help prevent commercial aircraft from being taken under fire, as in the Iranian Airbus incident of June 1988.

Non-cooperative target recognition (NCTR) has been pursued as a means of achieving positive identification of potential targets beyond visual range. One scheme explored has been the identification of targets based on their complex natural resonances. If a filter bank could be manufactured containing filters that would cancel these resonances for specific targets and the output energy of each filter compared to determine the lowest energy (each filter would minimize the output energy for the resonances received from the target for which it was constructed), then a target could be positively identified on the basis of these resonances alone.

Research efforts concerning the feasibility of using complex natural resonance cancellation have been conducted at the Naval Postgraduate School under the sponsorship of the Office of Naval Research (ONR) and the Defense Advanced Research Projects Agency (DARPA), and at Michigan State University under the sponsorship of the Naval Air Systems Command and DARPA. Each institution has developed a different approach to this problem, with M.A. Morgan of the Naval Postgraduate School developing a difference equation based K-Pulse technique, and K.M. Chen of Michigan State employing an integral equation based E-Pulse method.

This thesis will expand upon the work of Jean [Ref. 1] and Dunavin [Ref. 2] in further developing the K-Pulse concept. Additionally, the accuracy and reliability of the K-Pulse and E- Pulse techniques in performing the task of identifying target signatures based upon complex natural resonance cancellation will be compared.

II. BACKGROUND AND THEORY

A. BACKGROUND

Mains and Moffat [Ref. 3] developed the concept of extracting the natural resonant frequencies of a target from the transient response. Pioneering work in the area of natural resonances was performed by Baum at the Air Force Weapons Laboratory [Ref. 4]. Baum's technique, known as the Singularity Expansion Method (SEM), was derived for analysis of nuclear electromagnetic pulse effects (EMP) on strategic systems. The SEM concept embodies the result that system response may be represented as a weighted expansion of complex natural modes, at least to a certain extent. These modes are self-sustaining, but do decay, in the absence of excitation and are functions of the structural composition and geometry of the system. Additionally, the possible natural resonant modes are independent of the incident excitation, including angle of arrival and signal polarization. In the time domain, the modes are represented as exponentially decaying sinusoids. In the Laplace domain, they are expressed as complex functions composed of conjugate pole pairs in the left half of the s-plane. In radar applications, if a target is illuminated with RF energy, its natural modes have the potential to be excited to varying degrees and can then be identified. [Ref. 1]

Morgan [Ref. 5], Choong [Ref. 6], Manhila [Ref. 7], and Davenport [Ref. 8] have shown Baum's work to be incomplete in the case of scattered signals. The SEM pole-series expansion properly describes the signal only in the "late-time", which is defined as the time after the incident field has completed

illuminating the target. The "early-time" response includes scattered fields due to currents directly induced by incident field illumination, and cannot be used in extracting aspect independent natural resonances. [Ref. 5] The distinction between early-time and late-time responses is depicted in Figure 1.

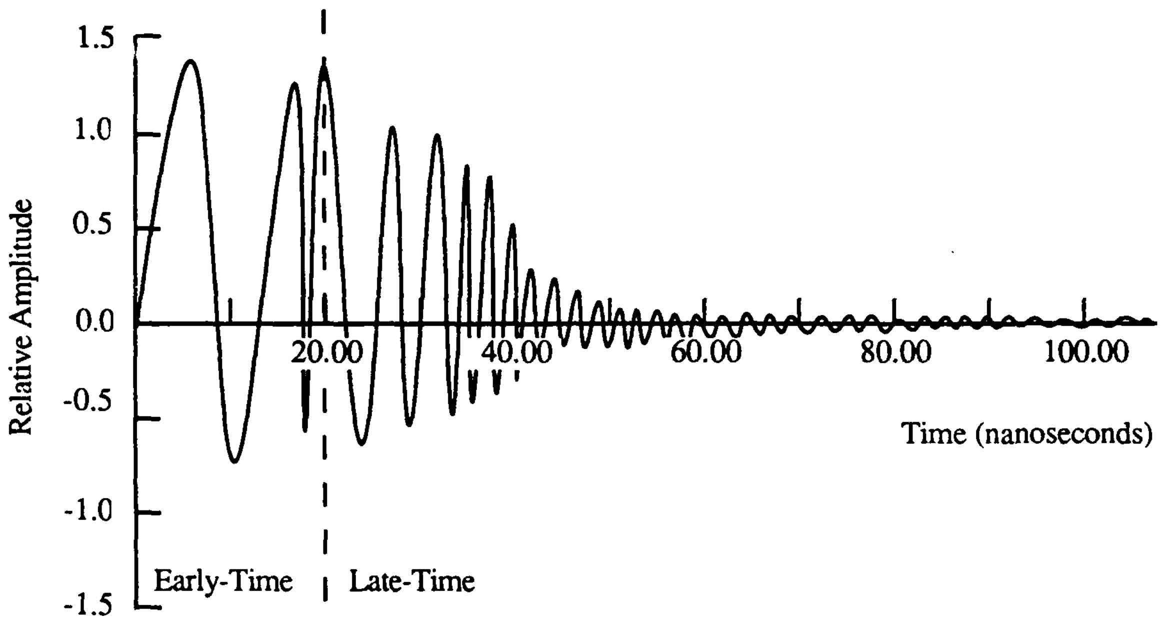


Figure 1. Typical Radar Return Signal

Preliminary research in NCTR was concentrated on target identification by pole extraction. These efforts exhibited extreme sensitivity to noise in the pole extraction algorithms. In addition, the processing of the complete signal response, rather than only the late-time portion was the norm in these early studies. Finally, the lengthy computing time required rendered pole extraction impractical in many instances for potential real-time identification systems. [Ref. 5]

The development of resonance annihilation filtering (RAF) NCTR schemes was first proposed by Kennaugh [Ref. 9] and has been furthered by Morgan [Refs. 5,10] and Chen [Ref. 11], based upon the potential of obtaining identifications in environments with signal to noise ratios (SNR's) on the order of 10 dB, combined with rapid signal processing times.

B. NATURAL RESONANCE SCATTERING AND THE RESONANCE ANNIHILATION FILTERING CONCEPT

When an electromagnetic wave is incident on a perfectly conducting body, the scattered energy will consist of the early-time response, induced by the initial excitation of the body, and the late-time, or natural mode response. The natural mode currents have been found to have a sinusoidal time variation which decays exponentially. These natural mode currents are related to the complex natural frequencies associated with the target body. The extent of individual mode excitation is dependent on the angle of incidence, polarization, and temporal behavior of the wave producing the excitation. However, the modes themselves are an inherent feature of the target body. This is what gives rise to the possibility of aspect independent target identification.

Due to the fact that the late-time signal components are exponentially decaying sinusoids, it follows that the poles must be located in the left half of the s-plane in the Laplace domain. Since the signal is a real quantity, the poles must occur as complex conjugate pairs. Additionally, as the currents eventually decay to zero, the poles will be located off the imaginary axis and will have an approximate linear relationship with each other. [Ref. 1] A

rigorous mathematical derivation of the scattered response has been demonstrated by Morgan [Ref. 5].

Considering both the early-time and late-time responses, the transient scattered signal may be represented by the following equation:

$$y(t) = y_E(t)[u(t) - u(t-T_0)] + y_L(t)u(t-T_0) + N(t) \tag{2.1}$$

Early-Time Late-Time Noise/Clutter

The changeover from early-time to late-time signal components is indicated by the use of unit step functions. The transition time T_0 is given as

$$T_0 = T + 2D/c \text{ seconds} \tag{2.2}$$

where

T = incident pulse width

D = line of sight dimension of scatterer

c = velocity of propagation

The late-time response is expressed as the sum of exponentially damped sinusoids:

$$y_L(t) = \sum_{n=1}^{\infty} A_n e^{s_n t} \cos(\omega_n t + \phi_n) \tag{2.3}$$

where

σ_n = damping coefficient of nth mode

ω_n = angular frequency of nth mode

A_n = amplitude of nth mode

ϕ_n = phase angle of nth mode

Noise and clutter are clearly unwanted signal components that complicate NCTR. [Ref. 5] Techniques for dealing with the noise components will be discussed as part of the E-Pulse and K-Pulse processes.

The RAF concept is illustrated in Figure 2. The output of the m-th filter is the convolution of its normalized impulse response $h_m(t)$ with the normalized input signal,

$$Z_m(t) = h_m(t)^* \left\{ y_E(t) - \left[u(t-T_0)\right] + y_L(t)u(t-T_0) + N(t) \right\} \qquad (2.4)$$

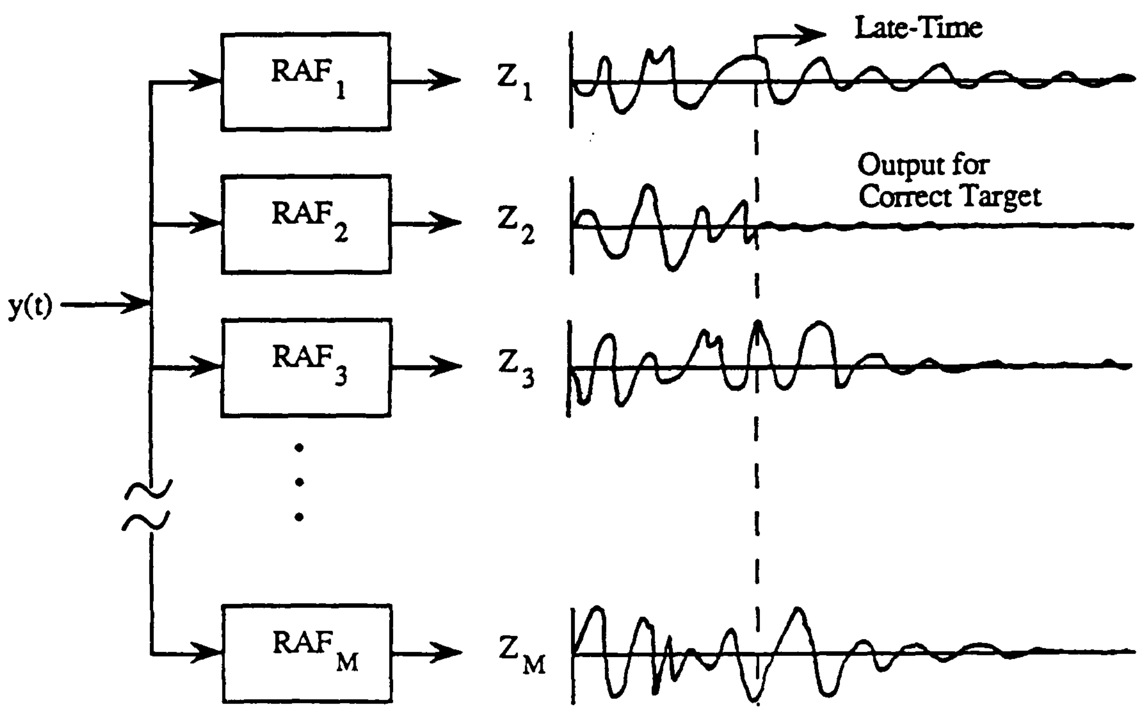


Figure 2. RAF Identification Concept [Ref. 5]

The input signal is normalized by its late-time RMS value,

$$E_y = \left[\sum_{k=1}^{N_y} y_L^2(t_k) \right]^{\frac{1}{2}} \qquad (2.4a)$$

and the impulse response is normalized by its total RMS value,

$$E_h = \left[\sum_{k=1}^{N_h} h_m^2(t_k) \right]^{\frac{1}{2}} \tag{2.4b}$$

where N_y and N_h are the respective number of time-series points in the late-time signal and the RAF filter impulse response.

The decision process for NCTR is based on selecting the target with the lowest signal energy in the late-time,

$$\varepsilon = \int_{T_L}^{T_R} Z_m^2(t)dt \tag{2.5}$$

T_L must be large enough to exclude significant energy from the early-time convolution output and T_R indicates record length.

Since the RAF concept is predicated on the cancellation of the natural modes of the target which are found in pure form only in the late-time, it is essential to prevent contributions to ε from the early-time signal. One way to accomplish this is to use nonrecursive finite impulse response (FIR) digital filters in RAF design, with impulse responses that are zero beyond a given time T_k such that $T_L > T_0 + T_k$. [Ref. 5]

Three key ingredients in RAF design are:

(1) The significant duration of individual RAF impulse responses, T_k, must be minimized. This will permit the convolution of the RAF impulse response and the early-time return to decay to zero as rapidly beyond T_0 as possible.

(2) The late-time energy for the matched target, when compared to that of other targets must be minimized over a wide range of aspect angles and polarizations.

(3) The RAF must be designed to minimize the transfer of the unwanted noise and clutter component, N(t), through the filter. [Ref. 5]

Criterion (1) will allow the maximum possible late-time interval for computing energy values, and will assist in optimizing the energy ratios in (2). However, the desire to minimize noise, which requires signal integration, conflicts with the desire to minimize the time for the convolution of the RAF impulse response and the early-time return signal to decay to near zero. Thus, there must be a tradeoff between the length of RAF impulse response duration and enhancement of noise and clutter suppression. [Ref. 5]

C. THE K-PULSE

The K-Pulse, as developed by Morgan at the Naval Postgraduate School, utilizes difference equations to form the RAF for natural mode cancellation. Recalling Equation 2.3, the late-time response is given as

$$y_L(t) = \sum_{n=1}^{\infty} A_n e^{\sigma_n t} \cos(w_n t + \phi_n)$$

For each individual mode, n, a_n and b_n may be found such that each individual mode is a solution to a three-point homogeneous difference equation of the form

$$a_n y[(p-1)\Delta t] + y[p\Delta t] + b_n y[(p+1)\Delta t] = 0 \tag{2.6}$$

where Δt is the sampling interval. Note that the solution is independent of the amplitude, A_n, and the phase angle, ϕ_n, which are aspect dependent.

Solving for a_n and b_n yields the following results:

$$a_n = -0.5 e^{\sigma_n \Delta t} / \cos(w_n \Delta t) \tag{2.7}$$

$$b_n = -0.5e^{-\sigma_n \Delta t} / \cos(w_n \Delta t) \tag{2.8}$$

To cancel N pole-pairs in the late-time response, a cascade of N digital 3-weight FIR filters can be employed. However, by combining coefficients that multiply identical sample points, a more simple FIR filter is generated with 2N delays and 2N + 1 weights. The difference equation representing this filter has the form

$$Z[p\Delta t] = \sum_{m=-N}^{N} C_m y[(p-m)\Delta t] \tag{2.9}$$

Although this is a non-causal filter, it can be used in real time by adding N time delays. In practice, use of this filter alone results in high frequency noise amplification, and requires smoothing of the input signal by low pass filtering. [Ref. 5]

Additionally, it has been shown by Dunavin [Ref. 2] that use of a Gaussian smoothing function is also effective in reducing the amplification of noise components in the input signal. A double Gaussian smoothing function has been incorporated in the construction of the K-Pulses in this thesis and will be explained in Chapter III.

In implementation of this technique for NCTR, K-Pulses for a variety of targets would be developed experimentally, stored in a computer library, and convolved with a target return. The output with the least amount of energy in the late-time will identify the correct target. [Ref. 2]

D. THE E-PULSE

The E-Pulse, as developed by Chen at Michigan State University, is also based on Equation 2.3, the expression for the late-time natural resonance response,

$$y_L(t) = \sum_{n=1}^{\infty} A_n e^{\sigma_n t} \cos(w_n t + \phi_n)$$

For a given signal, an E-Pulse of duration T_e can be constructed, which when convolved with a target signal, y(t) produces the return

$$Z(t) = \int_0^{T_e} e(t')y(t-t')dt' \tag{2.10}$$

where

Z(t) = the convolved output signal
e(t) = the E-Pulse [Ref. 11]

The E-Pulse is constructed of a set of basis functions, $f_k(t)$, which form a forcing component, r(t), and a calculated extinction component, x(t). Thus, e(t) may be represented as

$$e(t) = r(t) + x(t) = \sum_{k=1}^{K} \alpha_k f_k(t) \tag{2.11}$$

where α_k = amplitude of the k-th basis function. This is illustrated in Figure 3.

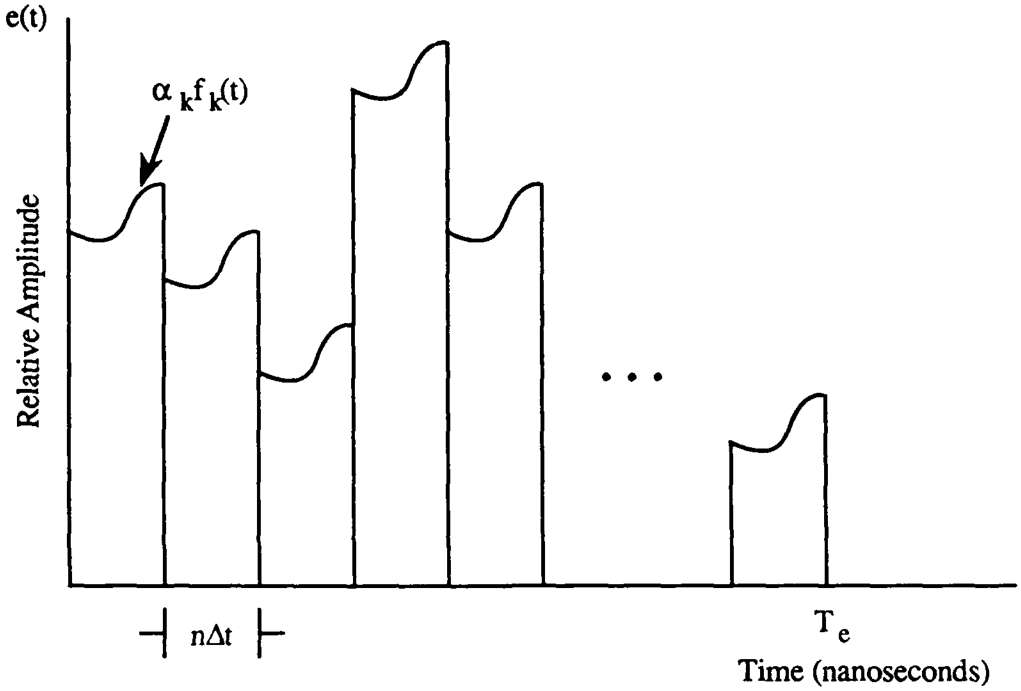


Figure 3. E-Pulse Composed of Basis Functions [Ref. 12]

Setting up the following equalities in the Laplace domain, and setting e(t) to zero,

$$\sum_{k=1}^{K} \alpha_k F_k(S_n) = -R(S_n) \tag{2.12}$$

$$\sum_{k=1}^{K} \alpha_k F_k(S_n^*) = -R(S_n^*) \tag{2.13}$$

where:

$$F_k(s) = \int_0^{Te} f_K(t) e^{-st} dt$$
$$R(s) = \int_0^{Te} r(t) e^{-st} dt$$

allows a solution to be found for the Laplace domain values of the basis functions. Using a matrix approach, and choosing k to form a square matrix yields, [Ref. 12]

$$\begin{bmatrix} F_1(S_1) & F_2(S_1) & \cdots & F_k(S_1) \\ \vdots & \vdots & & \vdots \\ F_1(S_n) & F_2(S_n) & \cdots & F_k(S_n) \\ F_1(S_1^*) & F_2(S_1^*) & \cdots & F_k(S_1^*) \\ \vdots & & & \vdots \\ F_1(S_n^*) & F_2(S_n^*) & \cdots & F_k(S_n^*) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_k \end{bmatrix} = \begin{bmatrix} -R(S_1) \\ \vdots \\ -R(S_n) \\ -R(S_1^*) \\ \vdots \\ -R(S_n^*) \end{bmatrix} \tag{2.14}$$

If identical basis functions are used for the forcing component as well as for the extinction component, these may be represented as time-shifted functions in the Laplace domain, as shown in the following matrix equation. The width of each basis function, given by Δt, is an integer multiple of the sampling interval used to sample the target signal. The use of basis functions

of widths that are integer multiples of the target signal sampling interval is effective in reducing the amplification of noise components in the input signal. A method for determining the optimum Δt will be presented in Chapter III.

$$
\begin{bmatrix}
F_0(S_1)e^{-s_1\Delta t} & F_0(S_1)e^{-s_1 2\Delta t} & \dots & F_0(S_1)e^{-s_1 k\Delta t} \\
\vdots & & & \vdots \\
F_0(S_n)e^{-s_n\Delta t} & F_0(S_n)e^{-s_n 2\Delta t} & \dots & F_0(S_n)e^{-s_n k\Delta t} \\
F_0(S_1^*)e^{-s_1^*\Delta t} & F_0(S_1^*)e^{-s_1^* 2\Delta t} & \dots & F_0(S_1^*)e^{-s_1^* k\Delta t} \\
\vdots & & & \vdots \\
F_0(S_n^*)e^{-s_n^*\Delta t} & F_0(S_n^*)e^{-s_n^* 2\Delta t} & \dots & F_0(S_n^*)e^{-s_n^* k\Delta t}
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\ \vdots \\ \alpha_n \\ \alpha_{n+1} \\ \vdots \\ \alpha_k
\end{bmatrix}
=
\begin{bmatrix}
-F_0(S_1) \\ \vdots \\ -F_0(S_n) \\ -F_0(S_1^*) \\ \vdots \\ -F_0(S_n^*)
\end{bmatrix}
\tag{2.14a}
$$

Each row may be divided by its common factor F_k, with the following result:

$$
\begin{bmatrix}
Q_1^{-1} & Q_1^{-2} & \dots & Q_1^{-k} \\
\vdots & \vdots & & \vdots \\
Q_n^{-1} & Q_n^{-2} & \dots & Q_1^{-k} \\
Q_1^{*-1} & Q_1^{*-2} & \dots & Q_1^{*-k} \\
\vdots & & & \vdots \\
Q_n^{*-1} & Q_n^{*-2} & \dots & Q_n^{*-k}
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\ \vdots \\ \alpha_n \\ \alpha_{n+1} \\ \vdots \\ \alpha_k
\end{bmatrix}
=
\begin{bmatrix}
-1 \\ \vdots \\ -1 \\ -1 \\ \vdots \\ -1
\end{bmatrix}
\tag{2.14b}
$$

where $Q_n^k = e^{s_n k\Delta t}$. This matrix equation may now be solved using the inverse of the matrix,

$$Q^{-1}\begin{bmatrix}-1\\ \vdots\\ \vdots\\ \vdots\\ -1\end{bmatrix}=\begin{bmatrix}\alpha_1\\ \vdots\\ \vdots\\ \vdots\\ \alpha_k\end{bmatrix} \tag{2.14c}$$

Note that the solutions for the basis function amplitudes are independent of phase angle and amplitude (aspect dependent attributes) found in the target signal, and are only a function of the damping coefficient and angular frequency (aspect independent) of the target response. [Ref. 12] This matrix approach has been utilized for all of the E-Pulse trials conducted for this thesis.

An alternate method of solving for the E-Pulse basis function amplitudes has been formulated that does not require the use of the forcing component r(t), [Ref. 11]. Substituting Equation 2.3 in Equation 2.10 for a signal response with N pole pairs in the late-time yields,

$$Z(t)=\sum_{n=1}^{N}A_n e^{\sigma_n t}\left[C_n\cos(\omega_n t+\phi_n)+D_n\sin(\omega_n t+\phi_n)\right] \tag{2.15}$$

where

$$C_n=\int_0^{T_e}e(t')\exp(-\sigma_n t')\cos(\omega_n t')dt' \tag{2.15a}$$

$$D_n=\int_0^{T_e}e(t')\exp(-\sigma_n t')\sin(\omega_n t')dt' \tag{2.15b}$$

Recalling the basis function construction of the E-Pulse and that k is chosen to equal 2N, C_n and D_n are formulated as follows,

$$C_n = \sum_{n=1}^{2N} M_{nk}^c \alpha_k \tag{2.16a}$$

$$D_n = \sum_{n=1}^{2N} M_{nk}^s \alpha_k \tag{2.16b}$$

where

$$M_{nk}^c = \int_0^{T_e} f_k(t')e^{-\sigma_n t'} \cos(\omega_n t')dt' \tag{2.16c}$$

$$M_{nk}^s = \int_0^{T_e} f_k(t')e^{-\sigma_n t'} \sin(\omega_n t')dt' \tag{2.16d}$$

Equations 2.16a and 2.16b may be converted to matrix form as follows,

$$\begin{bmatrix} C_n \\ \\ D_n \end{bmatrix} = \begin{bmatrix} M_{nk}^c \\ \\ M_{nk}^s \end{bmatrix} [\alpha_k] \tag{2.17}$$

where

n = 1, 2, ..., N

k = 1, 2, ..., 2N

and a solution for the basis function amplitudes obtained from the inverse of the matrix,

$$[\alpha_k] = \begin{bmatrix} M_{nk}^c \\ \\ M_{nk}^s \end{bmatrix}^{-1} \begin{bmatrix} C_n \\ \\ D_n \end{bmatrix} \tag{2.18}$$

To formulate an E-Pulse to annihilate the late-time signal response, all C_n and D_n are set to zero. In this event, $[\alpha_k]$ has a nontrivial solution only when

the determinant of $\begin{bmatrix} M_{nk}^{c} \\ M_{nk}^{s} \end{bmatrix}$ equals zero. This state will exist because all elements of the M matrix are functions of the duration of the E-Pulse, T_e. The required value of T_e is found numerically. Once the solution for T_e is obtained, $[\alpha_k]$ may be determined from a set of homogeneous equations constructed from Equation 2.17. [Ref. 11]

III. EXPERIMENTAL RESULTS

This chapter presents the results of experiments conducted using the K-Pulse and the E-Pulse to annihilate several different signals under varying conditions of noise. The testing of each annilhilation pulse was conducted in two phases. First, preliminary testing was conducted on the K-Pulse and the E-Pulse computer programs with synthetic signals used by Dunavin [Ref. 2] to demonstrate the effectiveness of the K-Pulse. Then, the K-Pulse and the E-Pulse were tested using high-Q, medium-Q, and low-Q synthetic target signals to determine their discrimination effectiveness against targets with only a five percent difference in pole composition, in environments with SNR's as low as 10 dB.

Section IIIA1 explains the use of the synthetic signal generator program, and Sections IIIA2 and IIIA3 illustrate the signals used for the preliminary testing and the follow-on discrimination effectiveness testing, respectively. Section IIIB describes the results of the K-Pulse preliminary and discrimination effectiveness tests and Section IIIC compiles these results for the E-Pulse.

A. THE SYNTHETIC SIGNALS

1. Synthetic Signal Generator Program

All data collected for this thesis was based on the computerized construction of synthetic waveforms. Since a target is characterized by its poles and zeros, a computer program can be utilized to manufacture a waveform that would serve as a return signal from a hypothetical target.

Such a program was developed by Jean [Ref. 1] in BASIC for the Tektronix 4052 microcomputer, and synthetic signals generated from this program were used by Dunavin [Ref. 2] in demonstrating the effectiveness of the K-Pulse. This program has been updated and converted to Microsoft FORTRAN for personal computer use in this thesis, and a program listing may be found in Appendix A.

One important feature of the program is a capability to generate both an early-time and a late-time portion of the signal. The early-time portion is modeled quite arbritrarily by the following equation:

$$\sin^2\left(\left[3\pi/2T\right] t\right) \tag{3.1}$$

where

T = early-time interval

Additionally, since the K-Pulse and E-Pulse techniques are implemented through digital signal processing, the synthetic signals must be sampled at or above the Nyquist frequency. As the late-time portion of the return signal is composed of exponentially decaying sinusoids, an algorithm has been included to allow the user to estimate an "effective" upper frequency content of the late-time signal spectrum. This upper frequency, f_c, is then used to bound the lower limit of sampling by $f_s \geq 2f_C$. To estimate f_s, the roll-off of the spectrum, F(f), above the highest spectral peak is compared to this peak, which is due to the highest order pole in the waveform. This is mathematically represented as

$$10\log_{10}\left|\frac{F(f_{\max})}{F(f_c)}\right|^2 = C \tag{3.2}$$

where

f_c = effective upper cutoff frequency of signal to be sampled

f_{max} = highest frequency of late-time poles

C = ratio of magnitudes in decibels

Considering only the highest order natural resonance mode, where $s_n = \sigma_{max} + j\omega_{max}$, the ratio of squared Fourier transforms becomes, with $\omega = 2\pi f$,

$$\left|\frac{F(f_{\max})}{F(f)}\right|^2 \approx \frac{(\omega - \omega_{\max})^2 + \sigma_{\max}^2}{\sigma_{\max}^2} \tag{3.2a}$$

Setting the dB value of this ratio to C, as per (3.2), the sampling frequency lower bound becomes

$$f_s \geq 2f_c \approx \frac{1}{\pi}\left\{\omega_{\max} + \sigma_{\max}\sqrt{10^{C/10} - 1}\right\} \tag{3.2b}$$

Noisy signals are produced using a Gaussian probability density function generator initially developed by Manhila [Ref. 7]. The algorithm allows the user to input a seed for generation of uniformly distributed random numbers, which are then converted to a Gaussian distribution. The proper signal to noise ratio (SNR) is then computed using average signal power in the late-time portion of the signal and the zero-mean white Gaussian noise is added to the synthetic signal.

2. Synthetic Signals Used in Preliminary Testing

The standard set of poles used by Dunavin [Ref. 2] was used to construct signals for preliminary testing of the K-Pulse and E-Pulse programs to determine if the pulses did indeed annihilate late-time signal responses, and is given in Table 1.

TABLE 1. STANDARD SET OF POLES[1]

n	POLES		RESIDUES	
	σ_n	ω_n	A_n	ϕ_n
1	-0.10	1.5	10	0
2	-0.19	2.85	7	$\pi/2$
3	-0.28	4.2	5	0
4	-0.39	5.85	3	π
5	-0.46	6.9	2	$\pi/2$
6	-0.54	8.1	1	0

These poles, as seen in the left half of the s-plane in the Laplace domain, are plotted in Figure 4.

Figure 5 is a time domain representation of the noiseless 6-pole waveform. The transition from the early-time to the late-time is clear. For comparison purposes, a 4-pole waveform, constructed of the first four poles listed in Table 1, is displayed in Figure 6. Frequency domain plots of the 6-pole waveform and the 4-pole waveform are provided in Figures 7 and 8, respectively. Examples of noisy waveforms are included for the 6-pole synthetic signal at SNR's of 30 dB (Figure 9), 20 dB (Figure 10), and 10 dB (Figure 11). A frequency domain representation of the 10 dB case is displayed in Figure 12, with the high frequency noise components clearly discernible. A 1-pole waveform consisting of the first pole listed in Table 1 and a 2-pole waveform consisting of the first two poles were also utilized in preliminary

---

[1]The following units apply to all pole data:

σ_n: G nep/sec — A_n: signal amplitude

w_n: G rad/sec — ϕ_n: radians

G: giga = 10^9

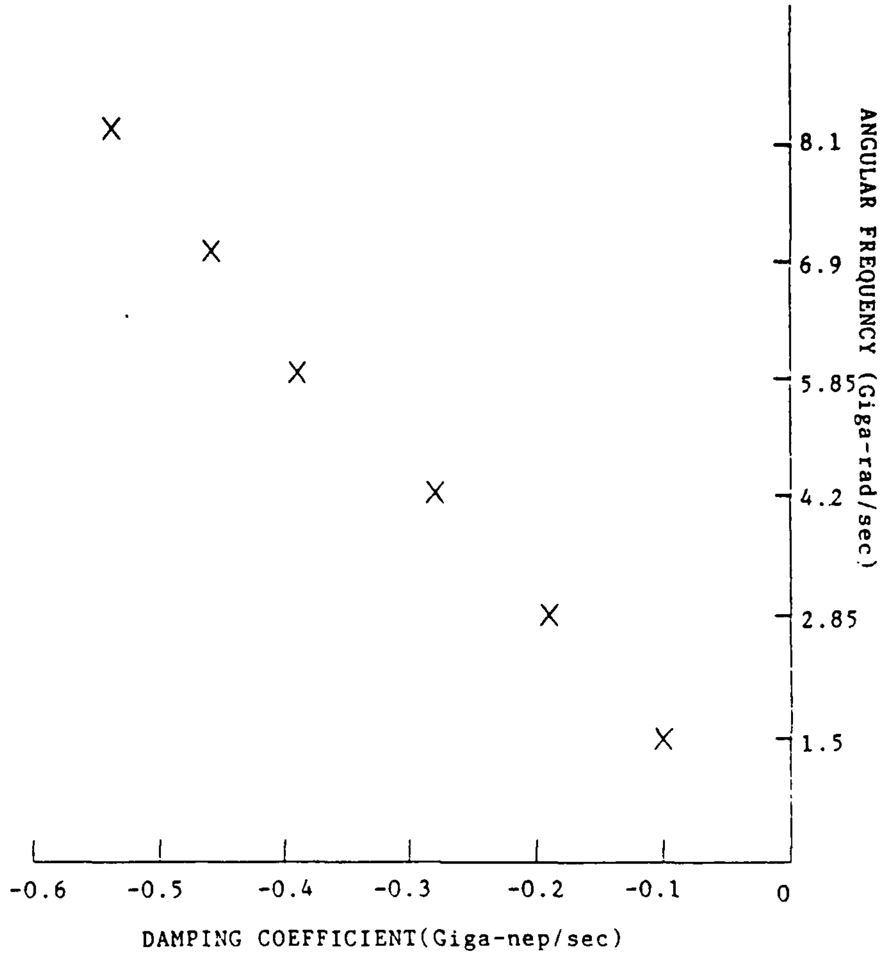


Figure 4. S-plane Plot of Standard Set of Poles for Preliminary Testing

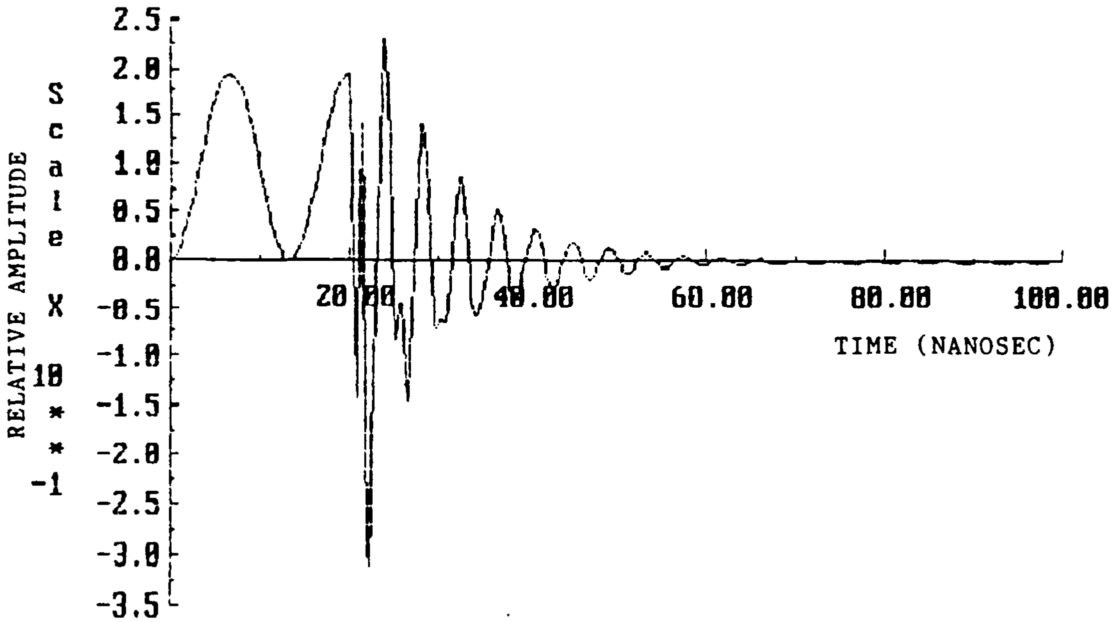


Figure 5. Time Domain Representation of Noiseless 6-Pole Waveform for Preliminary Testing

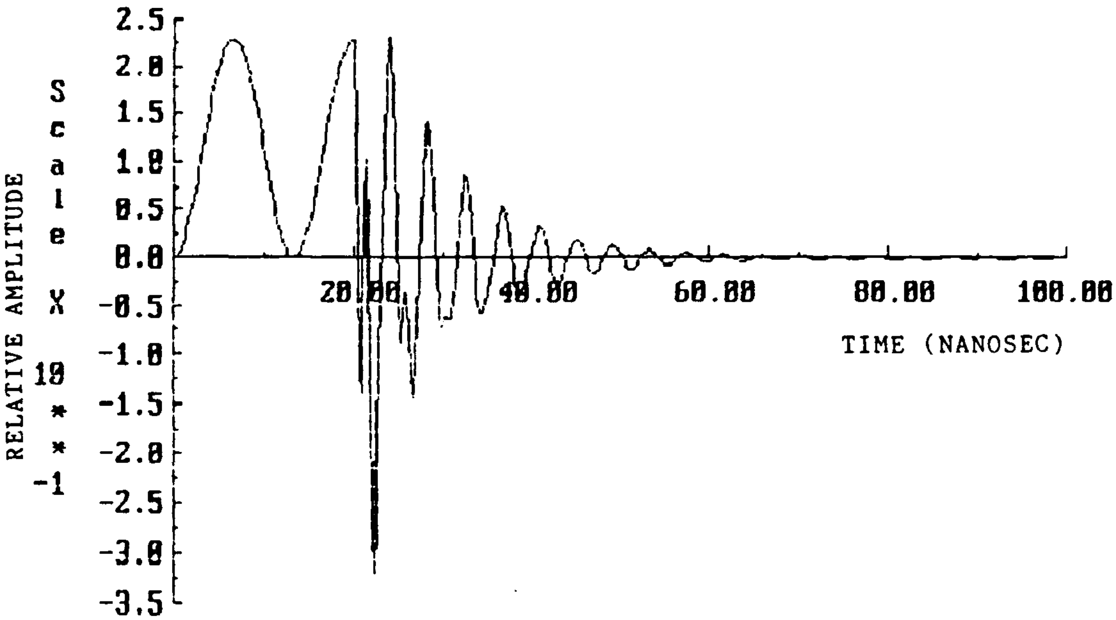


Figure 6. Time Domain Representation of Noiseless 4-Pole Waveform for Preliminary Testing

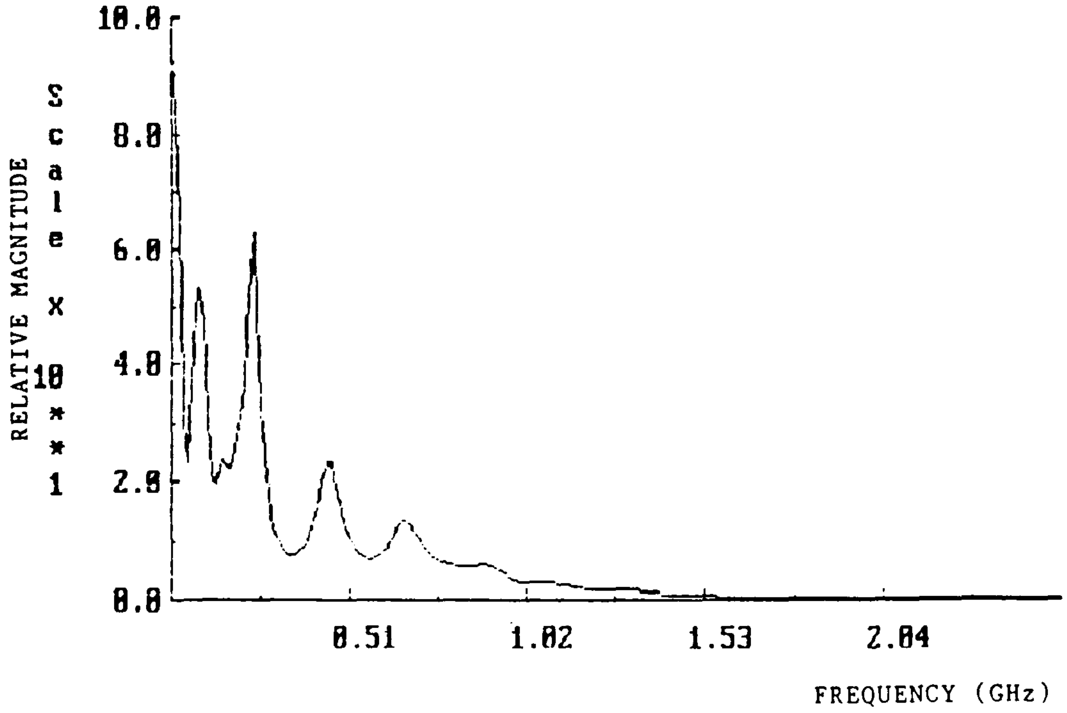


Figure 7. Frequency Spectrum of Noiseless 6-Pole Waveform for Preliminary Testing

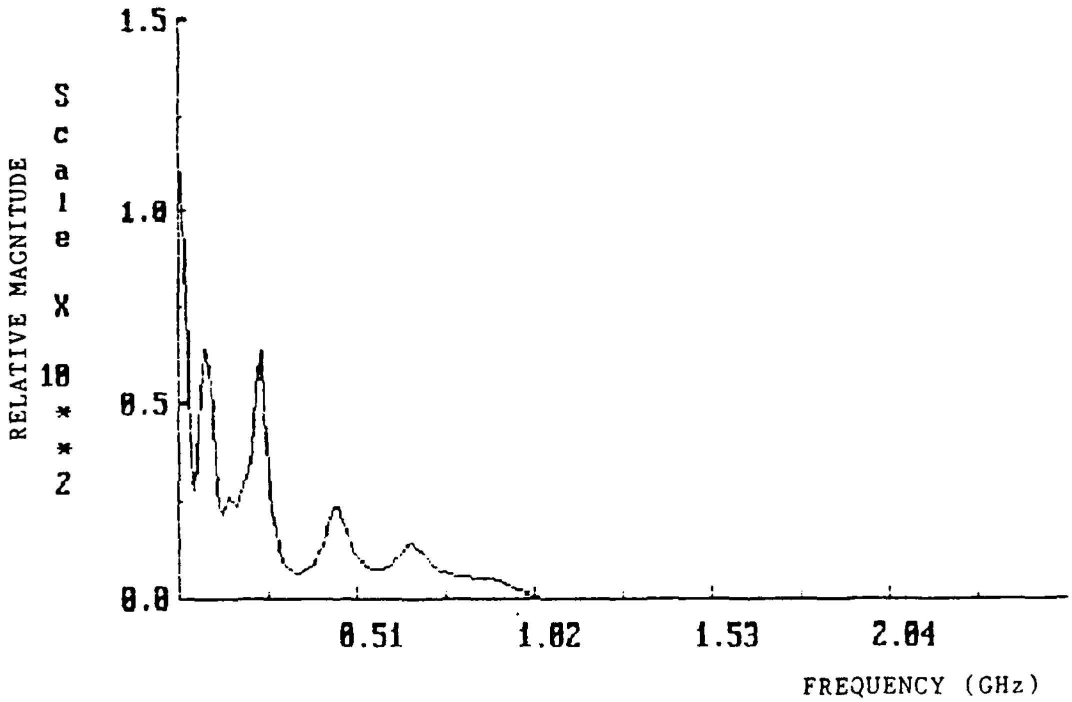


Figure 8. Frequency Spectrum of Noiseless 4-Pole Waveform for Preliminary Testing

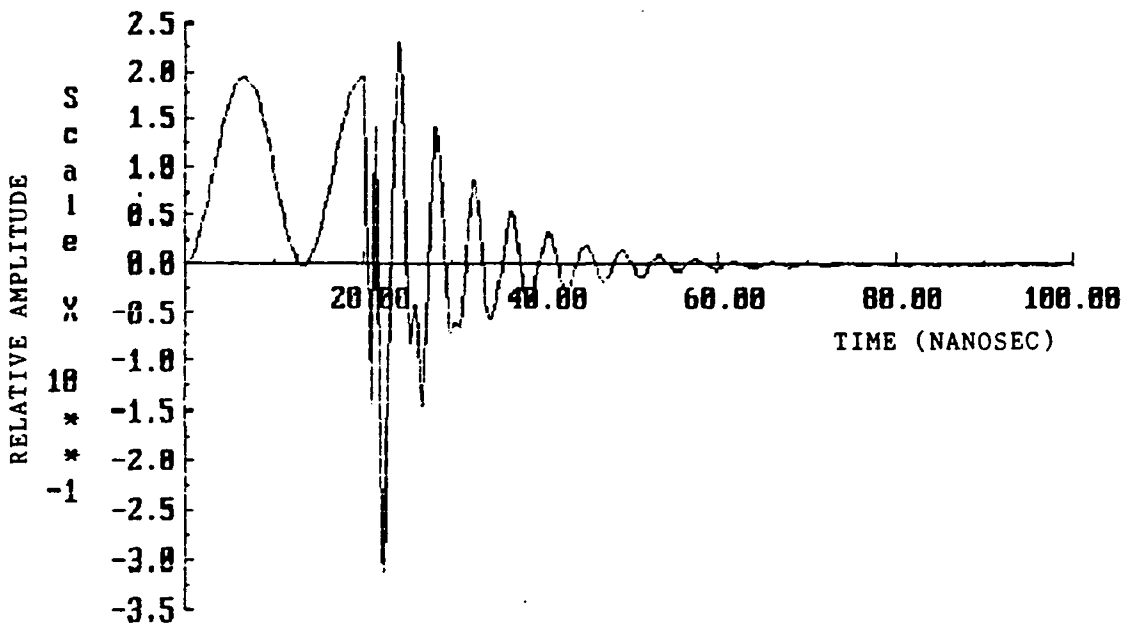


Figure 9. Time Domain Representation of 6-Pole Waveform for Preliminary Testing (SNR=30 dB)

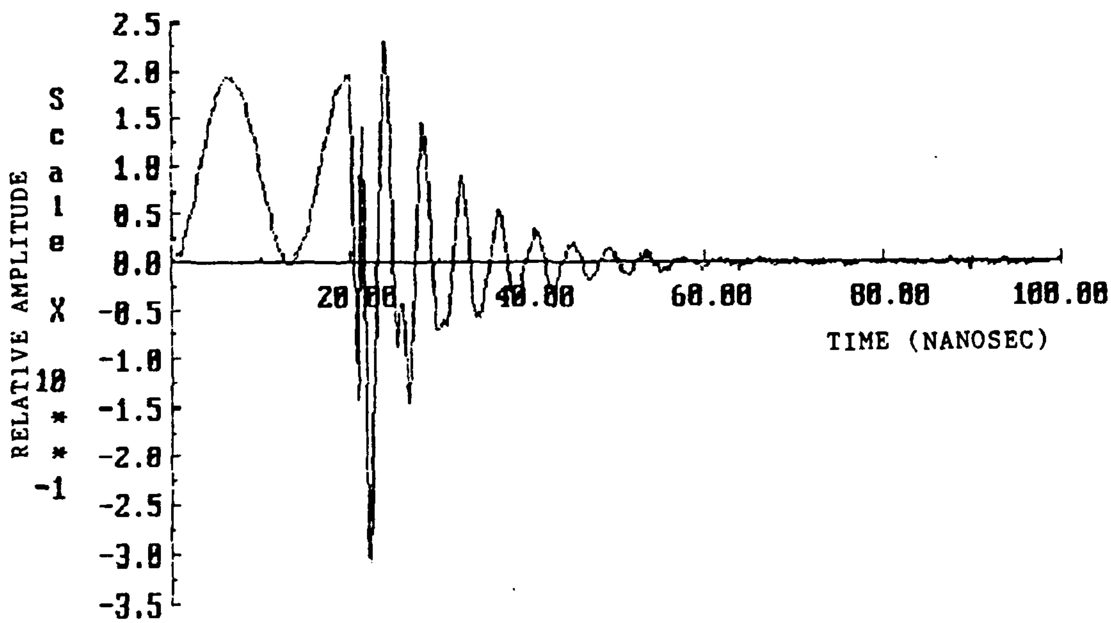


Figure 10. Time Domain Representation of 6-Pole Waveform for Preliminary Testing (SNR=20 dB)

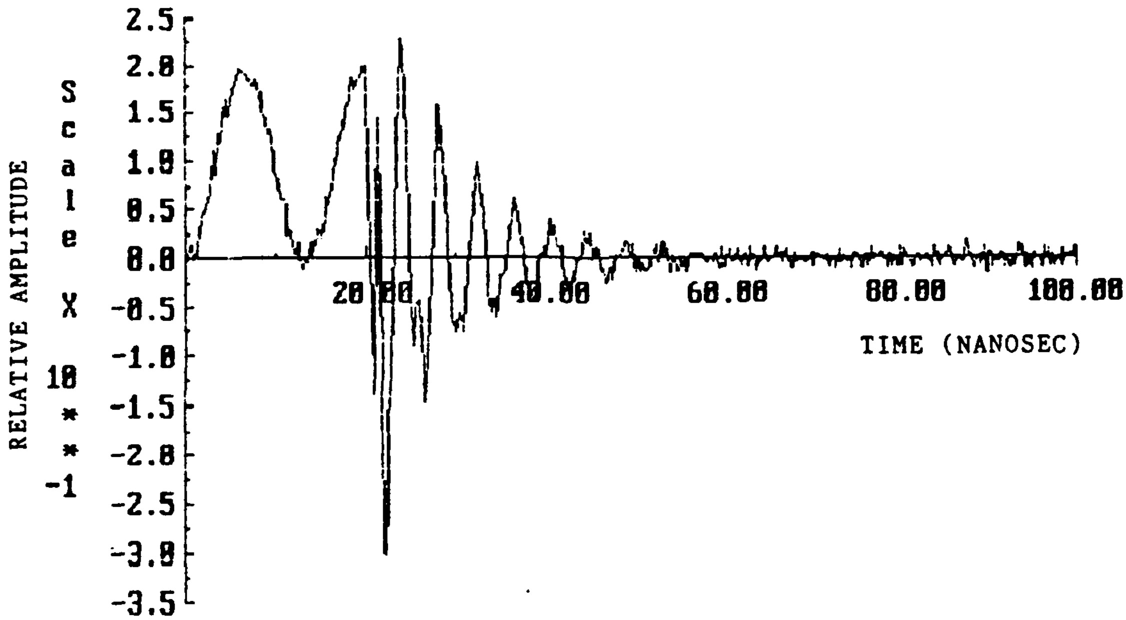


Figure 11. Time Domain Representation of 6-Pole Waveform for Preliminary Testing (SNR=10 dB)

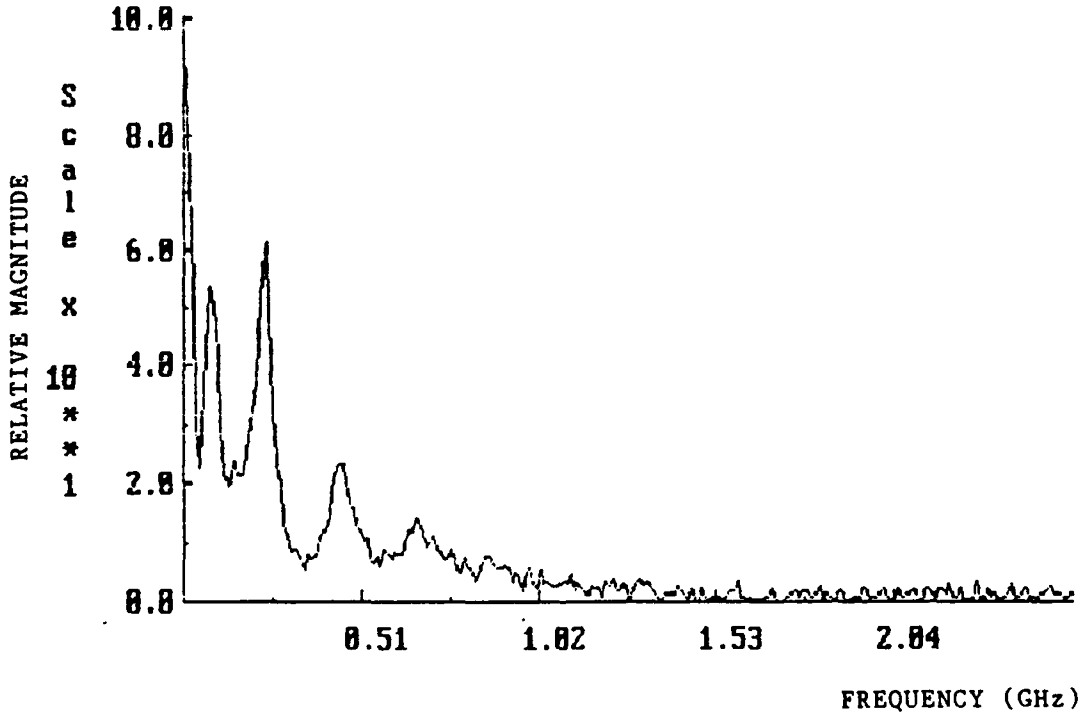


Figure 12. Frequency Spectrum of 6-Pole Waveform for Preliminary Testing (SNR=10 dB)

testing. Plots for these signals are not included for display, as they are similar to the four pole and six pole signals.

Table 2 is a matrix of sampling periods required for the four signals (derived from Equation 3.2), given a range of desired amounts of signal energy to be included beyond that of the spectral peak of the highest order pole. All signals used in the preliminary testing are 100 nanoseconds in length, with a 20 nanosecond early-time portion, and are sampled 512 times, for a sampling period of 0.1946 nanoseconds. Thus, each signal includes frequency components at least 20 dB below the highest order pole.

TABLE 2. SAMPLING PERIOD REQUIRED (NANOSECONDS) TO OBTAIN DESIRED SIGNAL LEVEL BELOW SPECTRAL PEAK OF HIGHEST ORDER POLE[2]

Number of Poles	Signal Level (dB)			
	0	–10	-20	–30
1	2.0944	1.7453	1.2591	0.6740
2	1.1023	0.9185	0.6627	0.3547
4	0.5370	0.4475	0.3208	0.1728
6	0.3878	0.3232	0.2331	0.1248

Figure 13 is the frequency domain representation of the noiseless 6-pole signal. The spectral peak of the highest order pole is indicated, and frequency components at a higher frequency than this peak up to a signal level approximately 25 dB below it are included.

---

[2] Based on pole characteristics from Table 1.

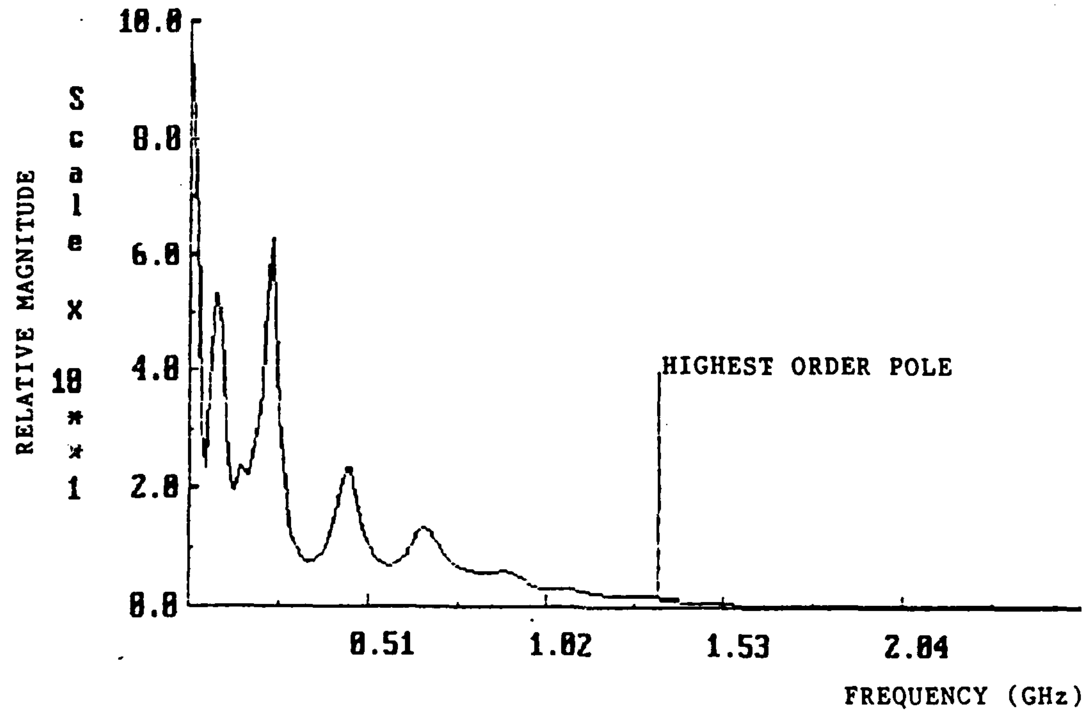


Figure 13. Frequency Spectrum of Noiseless 6-Pole Waveform with Highest Order Pole Indicated

3. Synthetic Signals Used in Discrimination Effectiveness Testing

Three signals were used to compare the effectiveness of the K-Pulse and the E-Pulse in discriminating highly damped (low-Q), moderately damped (medium-Q), and lightly damped (high-Q) signals in noiseless environments and in environments with SNR's of 30 dB, 20 dB, and 10 dB. The poles of each signal were each constructed using the same values for the corresponding ω's, as well as for the residues, in order to avoid the introduction of unwanted ambiguities into the test results. Each signal also had an identical early-time period for the same reason. The damping coefficient was determined by the following ratio,

$$\frac{\sigma_n}{\omega_n} = \frac{1}{2\pi}\left[\ln(k_n)\right] \tag{3.3}$$

where

$k_1 = 0.1$ (highly damped/low-Q target)
$k_2 = 0.5$ (moderately damped/medium-Q target)
$k_3 = 0.9$ (lightly damped/high-Q target)

This relationship is significant in representing a decay in the signal of $\exp(+\sigma_n \Delta t)=k$ in a single cycle, where $\Delta t = 2\pi/\omega_n$.

The poles used to construct the moderately damped/medium-Q waveform are given in Table 3, and their s-plane representation is included in Figure 14. The time domain and frequency domain plots for the noiseless case are displayed in Figure 15 and Figure 16, respectively.

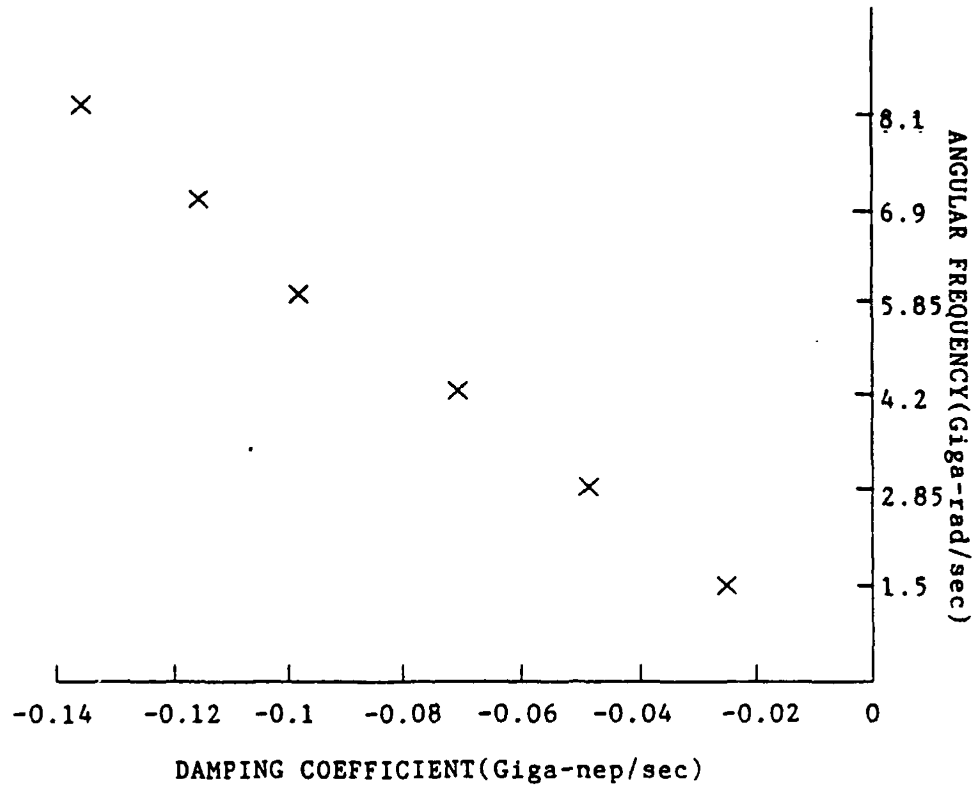


Figure 14. S-plane Plot of Lightly Damped/High-Q Waveform Poles

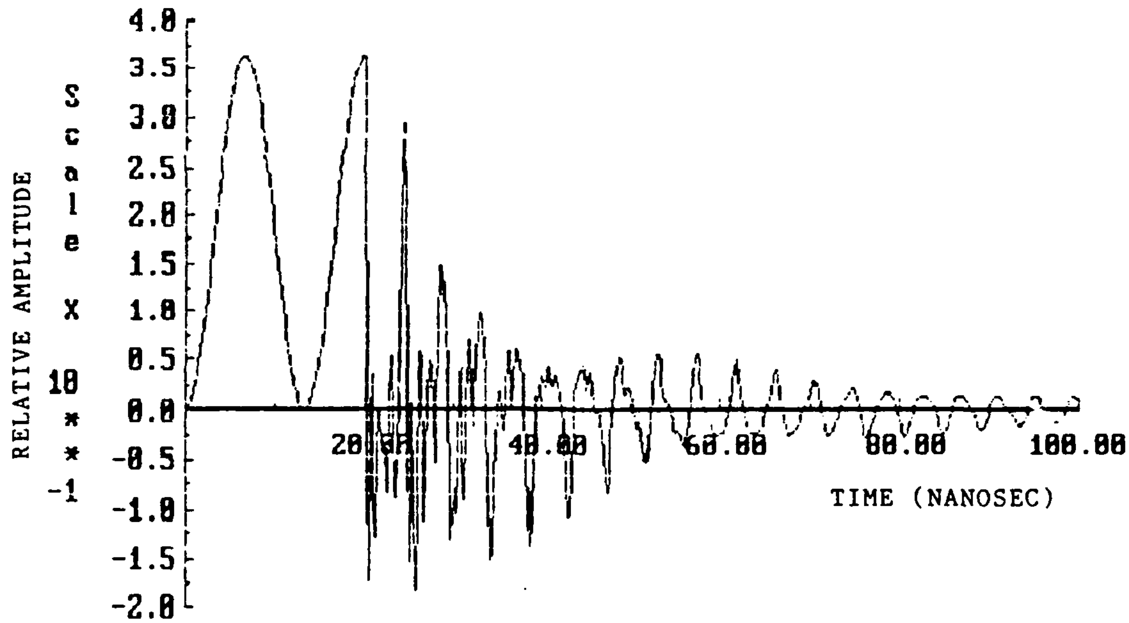


Figure 15. Time Domain Representation of Noiseless Lightly Damped/High-Q Waveform

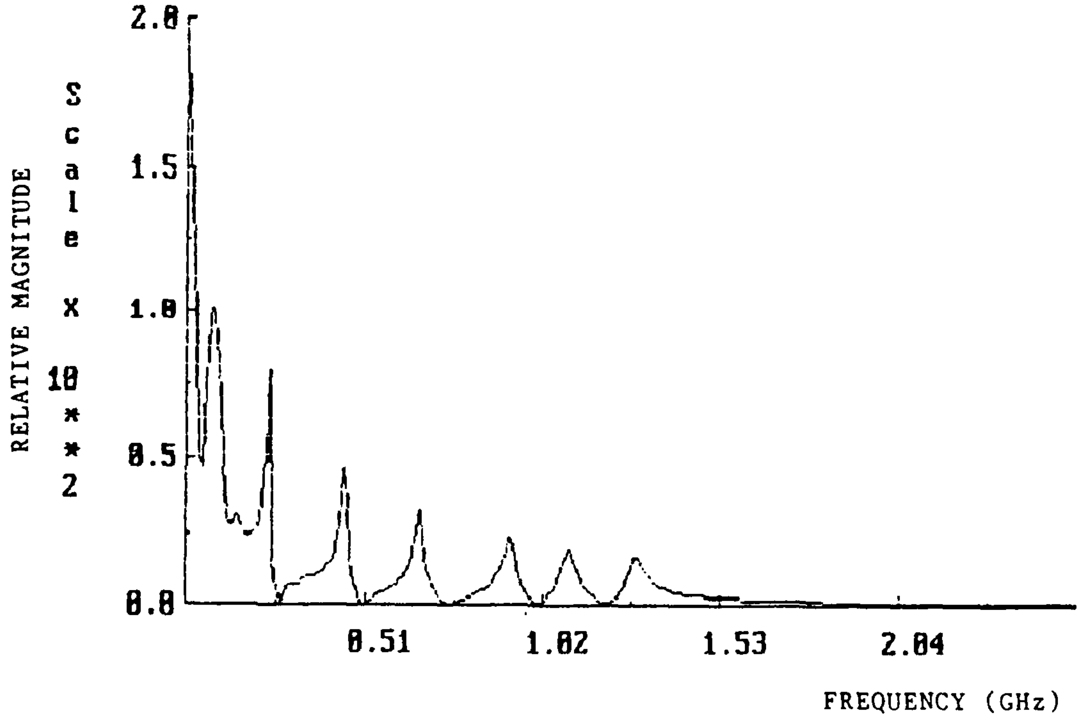


Figure 16. Frequency Spectrum of Noiseless Lightly Damped/High-Q Waveform

The poles used in constructing the moderately damped/medium-Q waveform are given in Table 4, and the s-plane plot is given in Figure 17. The noiseless time and frequency domain plots are displayed in Figures 18 and 19 respectively.

The poles used in constructing the highly damped/low-Q waveform are listed in Table 5, and their s-plane representation is plotted in Figure 20. The time and frequency domain plots for the noiseless case are displayed in Figures 21 and 22, respectively.

TABLE 3. LIGHTLY DAMPED/HIGH-Q WAVEFORM POLES

n	POLES		RESIDUES	
	σ_n	ω_n	A_n	ϕ_n
1	-0.0251	1.5	1	0
2	-0.0478	2.85	1	0
3	-0.0704	4.2	1	0
4	-0.0981	5.85	1	0
5	-0.1157	6.9	1	0
6	-0.1358	8.1	1	0

TABLE 4. MODERATELY DAMPED/MEDIUM-Q WAVEFORM POLES

n	POLES		RESIDUES	
	σ_n	ω_n	A_n	ϕ_n
1	-0.1655	1.5	1	0
2	-0.3144	2.85	1	0
3	-0.4633	4.2	1	0
4	-0.6454	5.85	1	0
5	-0.7612	6.9	1	0
6	-0.8936	8.1	1	0

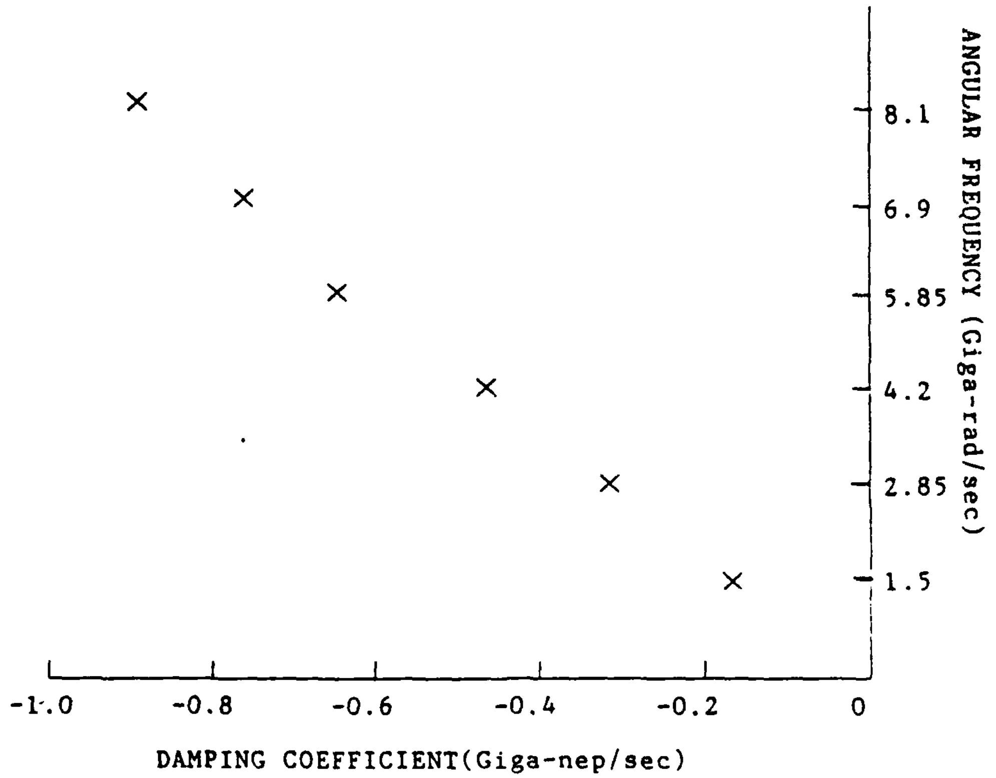


Figure 17. S-plane Plot of Moderately Damped/Medium-Q Waveform Poles

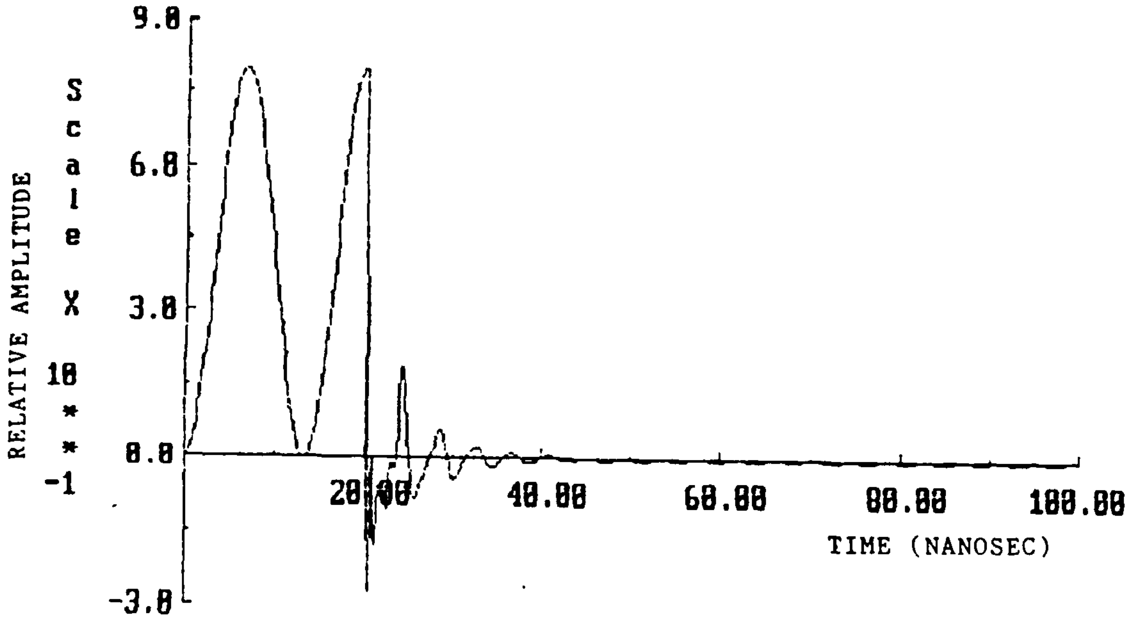


Figure 18. Time Domain Representation of Noiseless Moderately Damped/Medium-Q Waveform

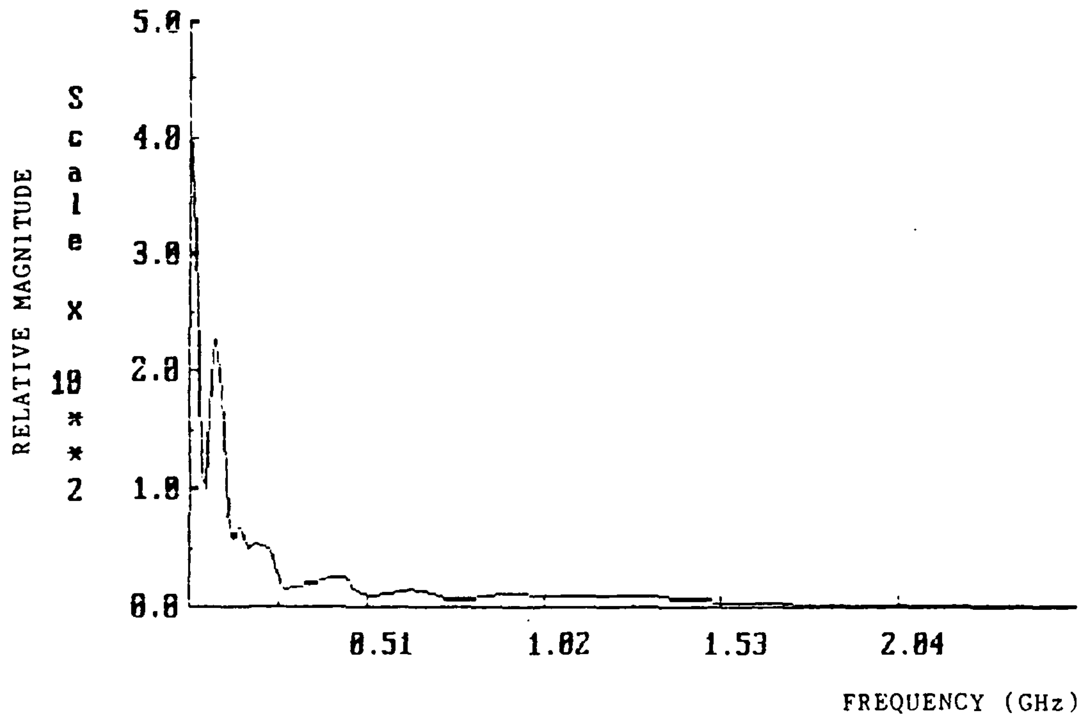


Figure 19. Frequency Spectrum of Noiseless Moderately Damped/Medium-Q Waveform

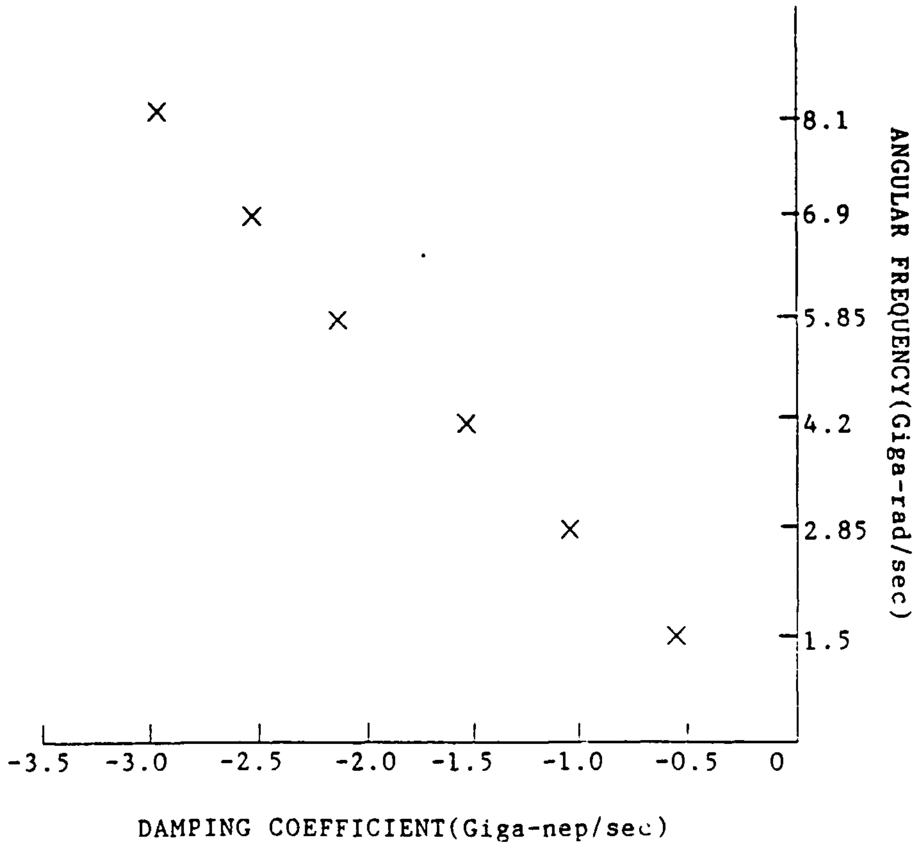


Figure 20. S-plane Plot of Highly Damped/Low-Q Waveform Poles

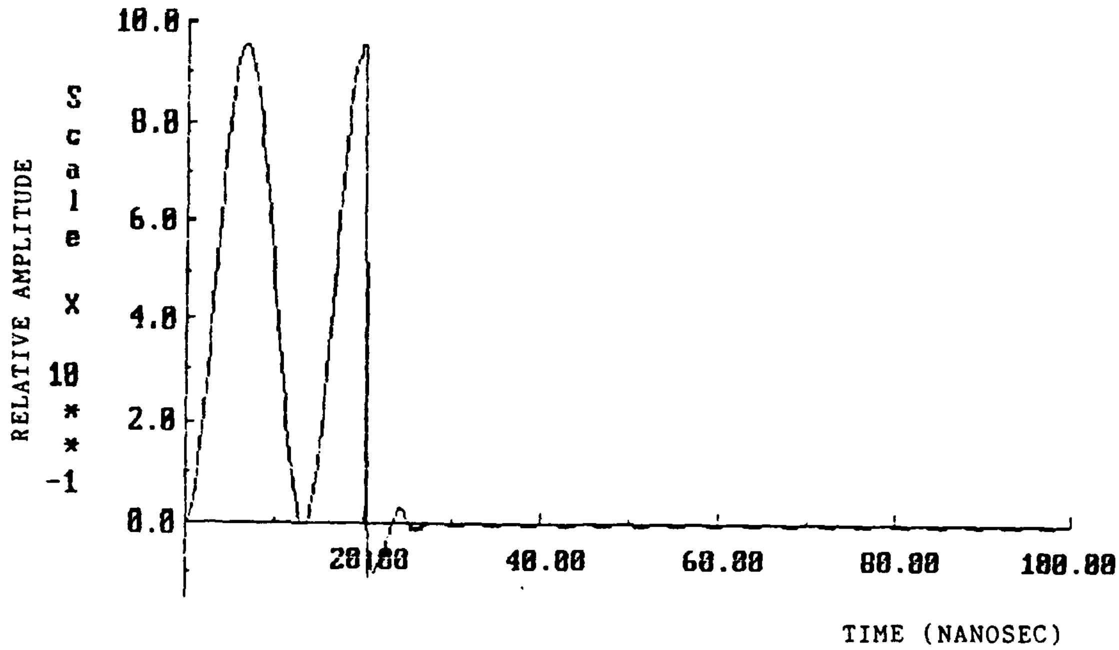


Figure 21. Time Domain Representation of Noiseless Highly Damped/Low-Q Waveform

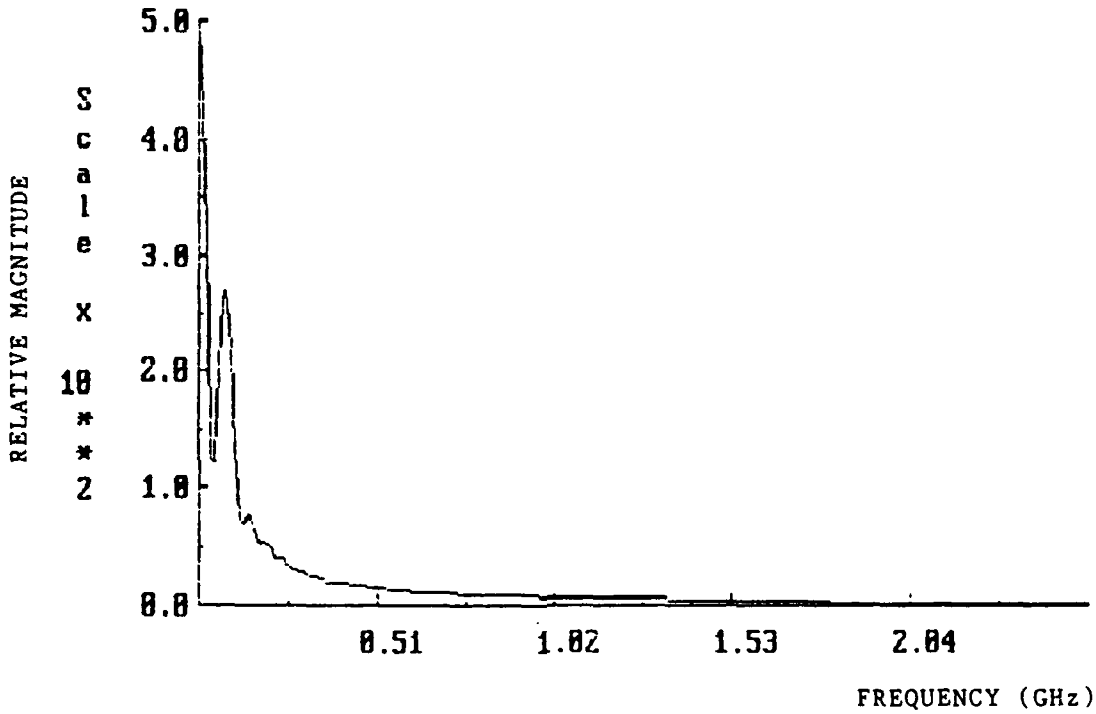


Figure 22. Frequency Spectrum of Noiseless Highly Damped/Low-Q Waveform

TABLE 5. HIGHLY DAMPED/LOW-Q WAVEFORM POLES

n	POLES		RESIDUES	
	σ_n	ω_n	A_n	ϕ_n
1	-0.5497	1.5	1	0
2	-1.0444	2.85	1	0
3	-1.5392	4.2	1	0
4	-2.1438	5.85	1	0
5	-2.5286	6.9	1	0
6	-2.9684	8.1	1	0

B. RESULTS OF K-PULSE TESTING

1. Algorithm Development

A Microsoft FORTRAN program implementing the simplified K-Pulse FIR filter of Equation 2.9, through the use of the z-transform polynomial, was developed and the program listing is included in Appendix B. Preliminary testing of the program was conducted using the 6-pole, 4-pole, 2-pole, and 1-pole signals described in Section IIIA2 above, in an effort to confirm Dunavin's results, which were obtained using the three-point operator K-Pulse approach of Equation 2.6. [Ref. 2] K-Pulses were created with filter weights spaced at intervals equal to the sampling period of the test signals.

a. Double Gaussian Smoothing Function

In an effort to enhance the capability of the K-Pulse to discriminate signals in noisy environments, Dunavin used both Gaussian and double Gaussian smoothing functions in forming K-Pulses. For this thesis, double Gaussian smoothing functions of varying widths were constructed to determine the optimum width for both late-time resonance

annihilation and noise reduction. A FORTRAN program listing for the smoothing function is included in Appendix C.

The double Gaussian smoothing function is composed of a "fast-acting" Gaussian function, and a "slow-acting" Gaussian function. When the slow-acting function is subtracted from the fast-acting function, the resulting double Gaussian function has a time domain representation as illustrated in Figure 23. Importantly, the double-Gaussian function has zero DC content. The width of the fast-acting Gaussian function (the portion above the x-axis) is controlled by the exponent used in its formation,

$$\alpha_1 = B\omega_{min} / 2[\ln 10]^{\frac{1}{2}} \qquad (3.4)$$

where

ω_{min} = lowest angular frequency in signal
B = spreading *coefficient*

As B is increased in magnitude, the smoothing function becomes narrower in the time domain. As the smoothing function narrows in the time domain, its frequency response becomes wider. Figure 24 is a time domain representation of a double Gaussian smoothing function with B=1.01. The frequency response is illustrated in Figure 25. For comparison purposes, Figure 26 is a time domain representation of a smoothing function with B=3.5. The frequency response is given in Figure 27.

To prevent lengthy signal processing times, the double Gaussian smoothing function is truncated at the minimum point where its essential frequency characteristics are preserved. Since the process of truncation is

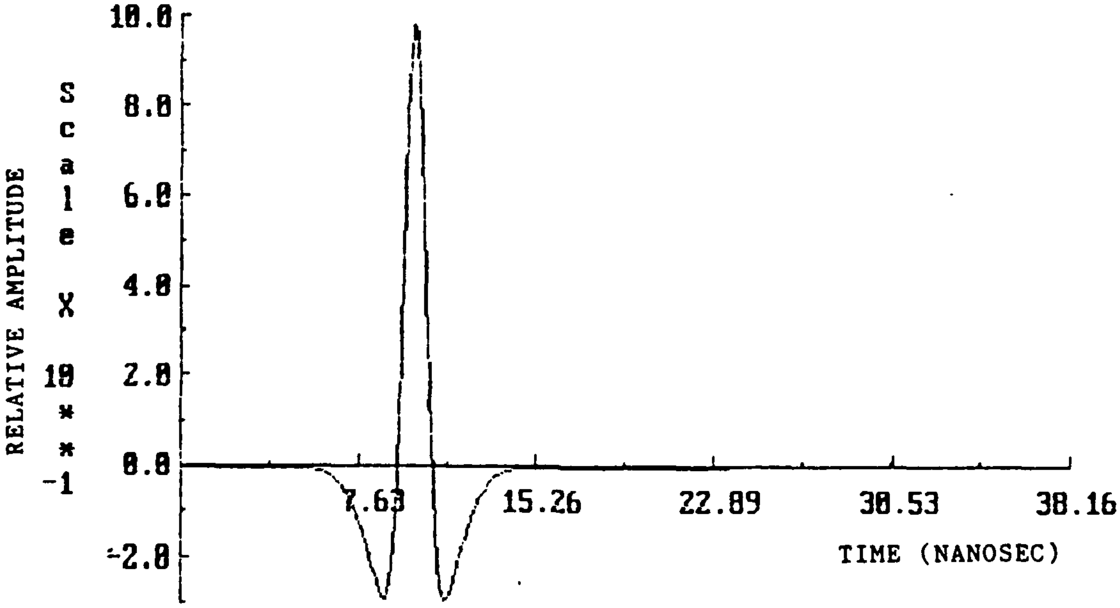


Figure 23. Representative Double Gaussian Smoothing Function

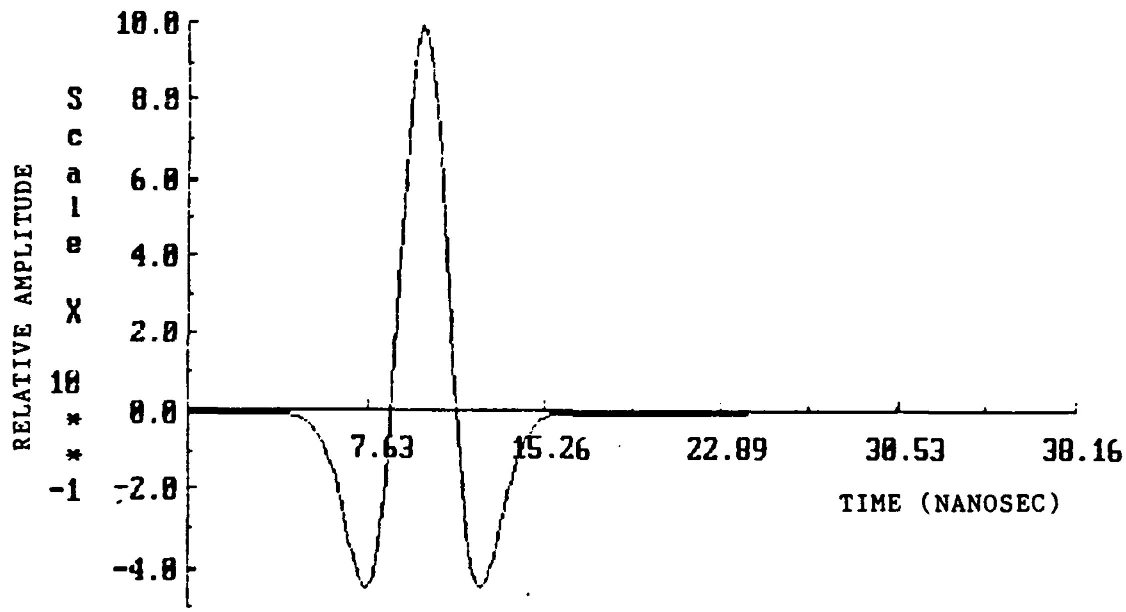


Figure 24. Time Domain Representation of Double Gaussian Smoothing Function (B=1.01)

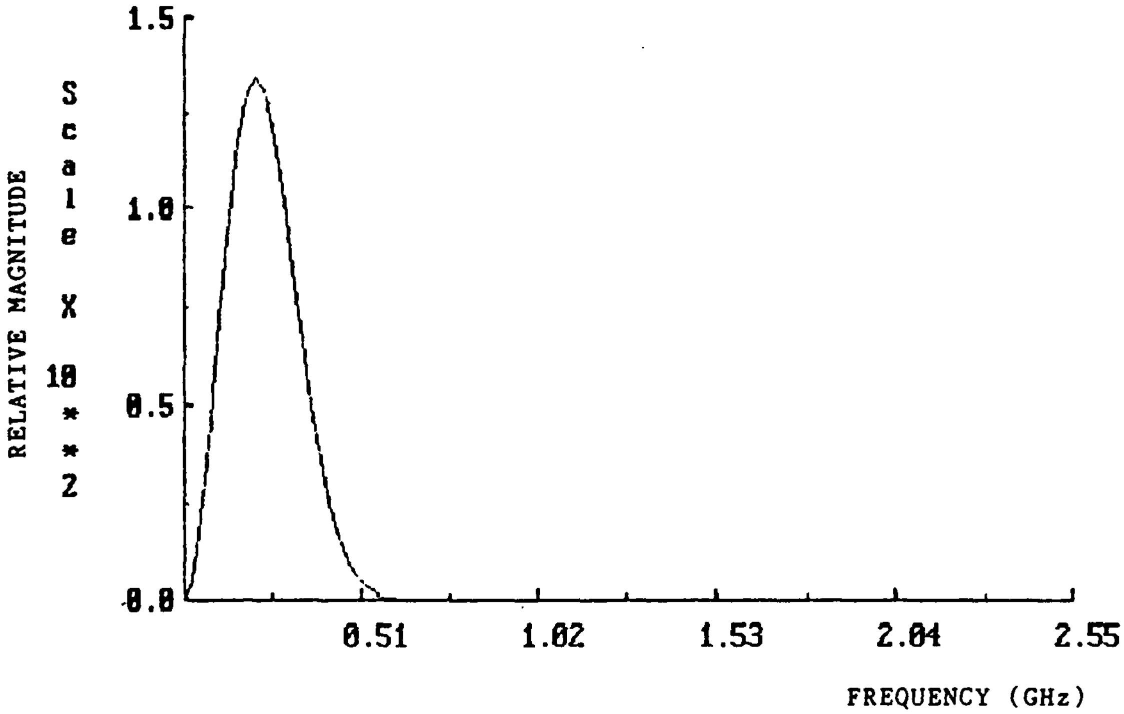


Figure 25. Frequency Response of Double Gaussian Smoothing Function (B=1.01)

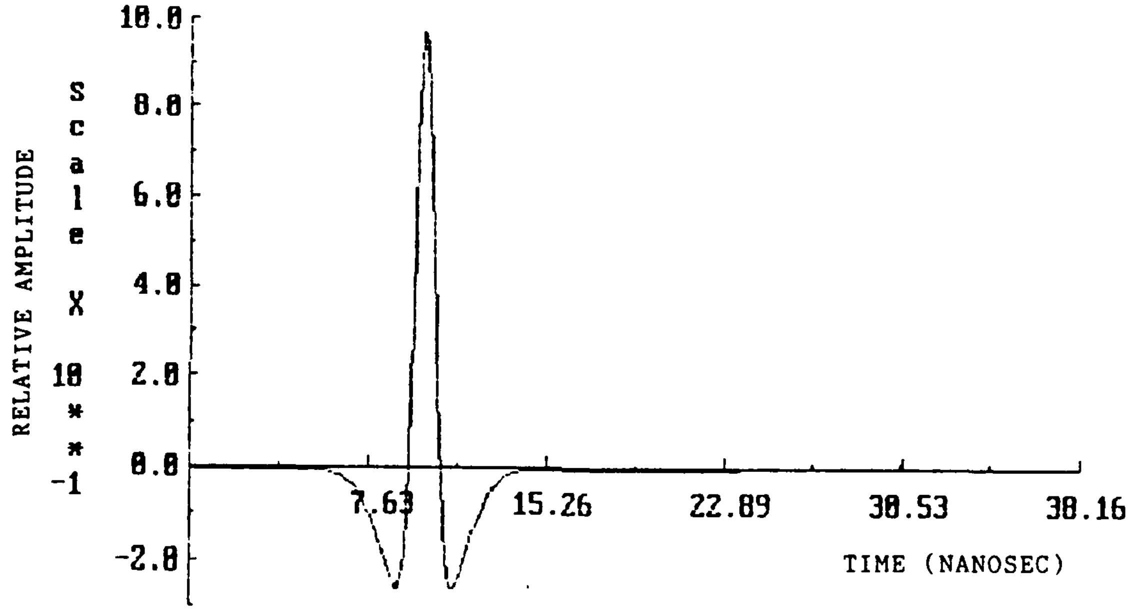


Figure 26. Time Domain Representation of Double Gaussian Smoothing Function (B=3.5)

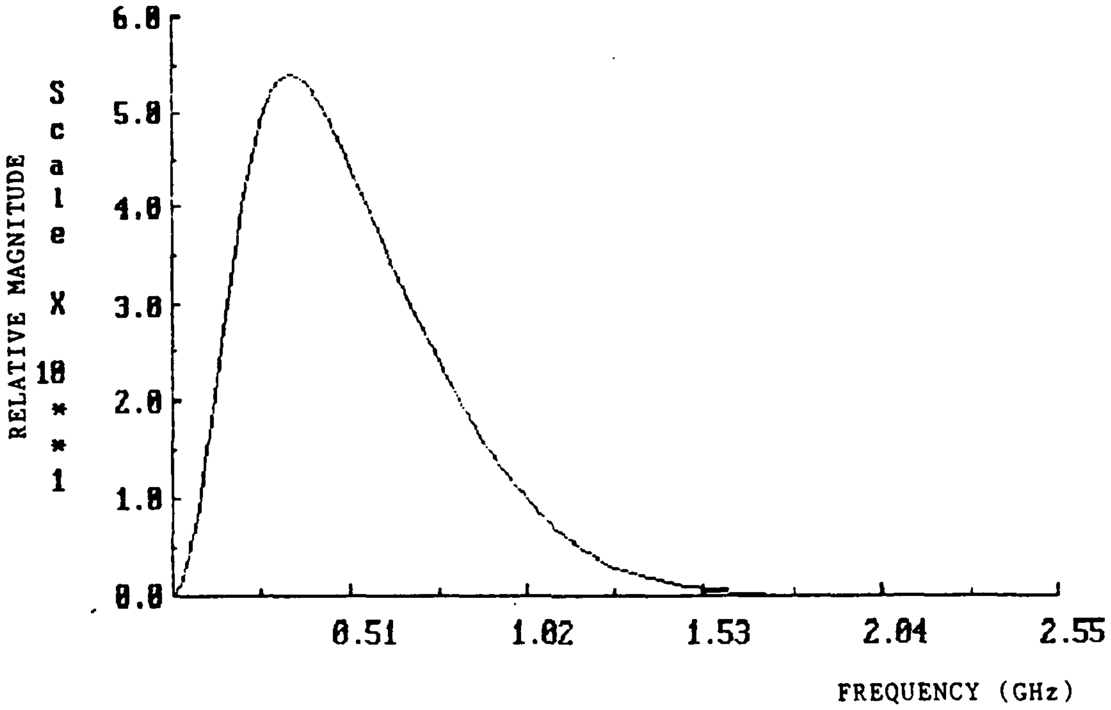


Figure 27. Frequency Response of Double Gaussian Smoothing Function (B=3.5)

mathematically equivalent to multiplying the smoothing function by a square window function in the time domain, the result in the frequency domain is a convolution of the smoothing function frequency response with a sinc function. This causes an undesired frequency response in the high frequency range. This problem is reduced by multiplying the truncated double Gaussian function with a Hamming window in the time domain.

Once the double Gaussian smoothing function has been formed, truncated, and smoothed with a Hamming window, it is convolved with the previously formed K-Pulse and normalized by its RMS energy to construct a smoothed K-Pulse. The program listing for this convolution routine is included in Appendix D.

2. Preliminary Testing

K-Pulses were created and smoothed with double Gaussian functions with spreading coefficients of 1.01, 2.0, 3.0, and 3.2 for the 1-pole, 2-pole, 4-pole, and 6-pole test signals. An unsmoothed K-Pulse was also constructed for each signal to establish the effectiveness of the K-Pulse concept. These K-Pulses were then convolved with the test signals to determine if late-time complex resonance annihilation could be achieved. The program listing for the FORTRAN implementation of this convolution routine is included in Appendix E.

The 6-pole unsmoothed K-Pulse created using the pole data found in Table 1 is displayed in Figure 28. The frequency spectrum is displayed in Figure 29. It is noted that the magnitude of the K-Pulse increases

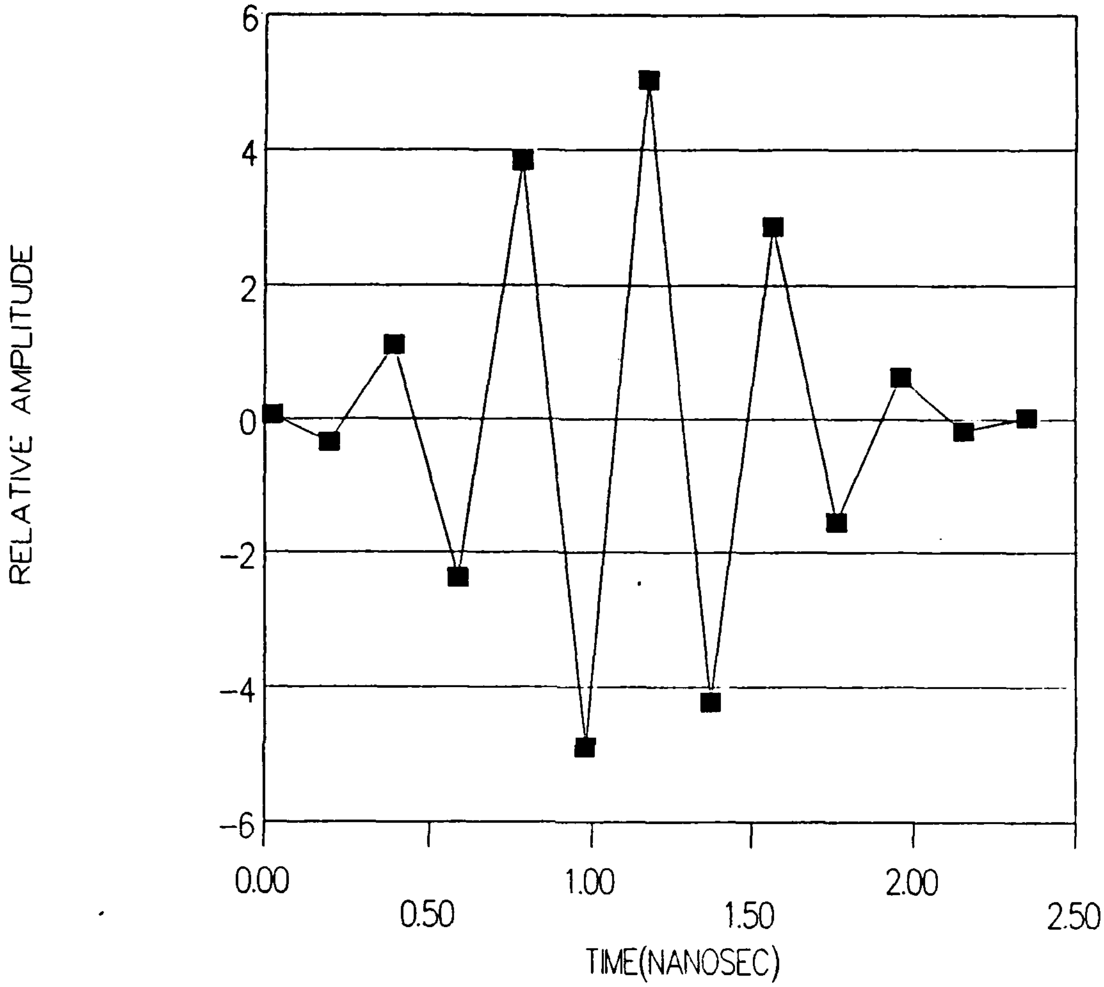


Figure 28. Time Domain Representation of Unsmoothed 6-Pole K-Pulse

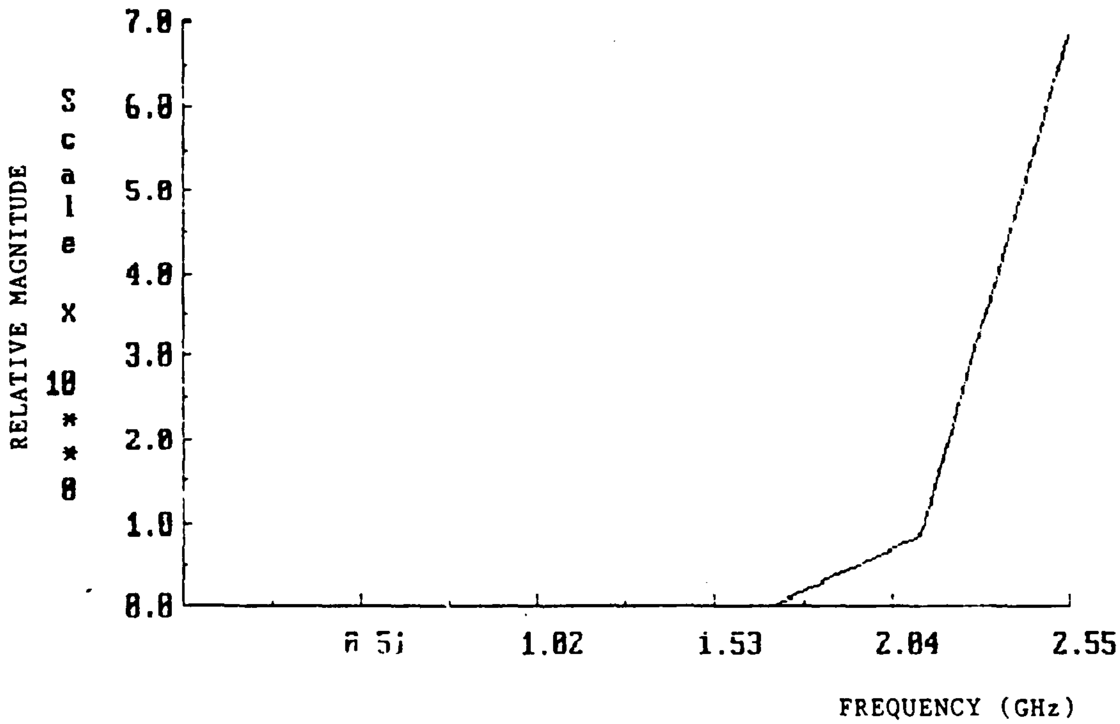


Figure 29. Frequency Response of Unsmoothed 6-Pole K-Pulse

monotonically with frequency, and will amplify any high frequency noise components present in a target return signal. The smoothed K-Pulse with a spreading coefficient of 1.01 is illustrated for comparison purposes in Figure 30, and the frequency spectrum is plotted in Figure 31.

The frequency spectra for the 6-pole smoothed K-Pulses with spreading coefficients of 2.0, 3.0, and 3.2 are displayed in Figures 32, 33, and 34 respectively. The spectrum of the K-Pulse smoothed with a double Gaussian function using a spreading coefficient of 3.0 (Figure 33) displays a highly developed structure, with deep nulls corresponding to the poles of the 6-pole signal from which it was constructed.

The spectra of the K-Pulses smoothed with functions using spreading coefficients 1.01 (Figure 31) and 2.0 (Figure 32) do not contain the completely developed structure of Figure 33, and do not show deep nulls for each specific signal pole. The spectrum for the K-Pulse using spreading coefficient 3.2 (Figure 34) has almost the same null structure as Figure 33, but has the undesired characteristic of a rising response at the high end of the frequency range. Plots for the 4-pole, 2-pole and 1-pole K-Pulses are not included due to their similarity to the 6-pole K-Pulses.

Results indicate that the unsmoothed K-Pulse produces the lowest energy output in the late-time for the 4-pole and 6-pole noiseless signals only. The unsmoothed K-Pulse will not minimize the late-time energy of the noisy signals because of the amplification of high frequency noise components inherent in its frequency spectrum (Figure 28). The smoothed K-Pulse with

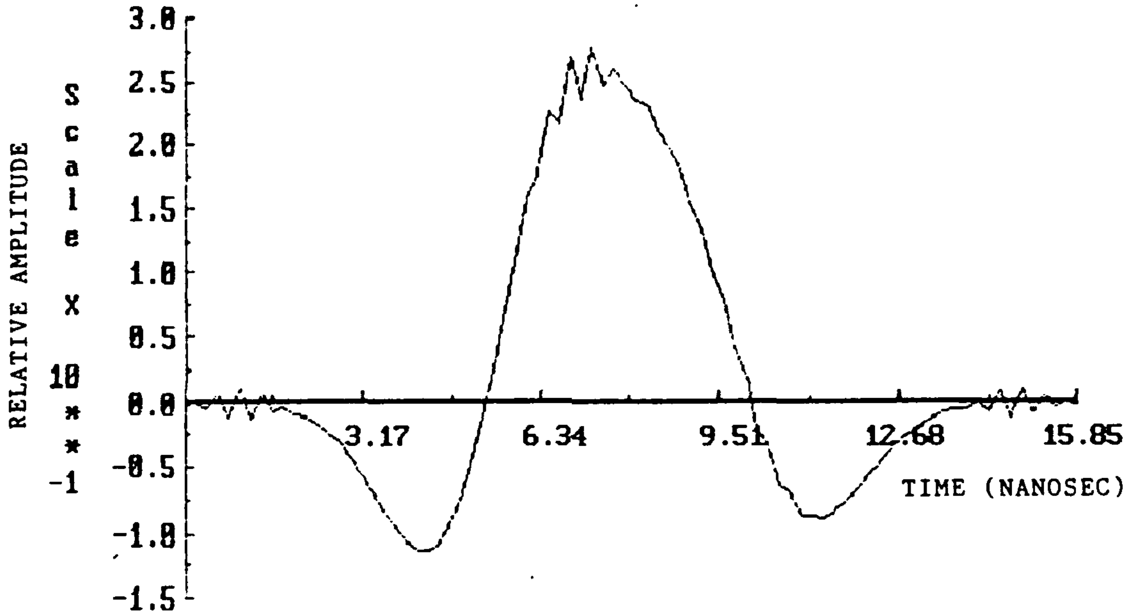


Figure 30. Time Domain Repres ntation of Smoothed 6-Pole K-Pulse (B=1.01)

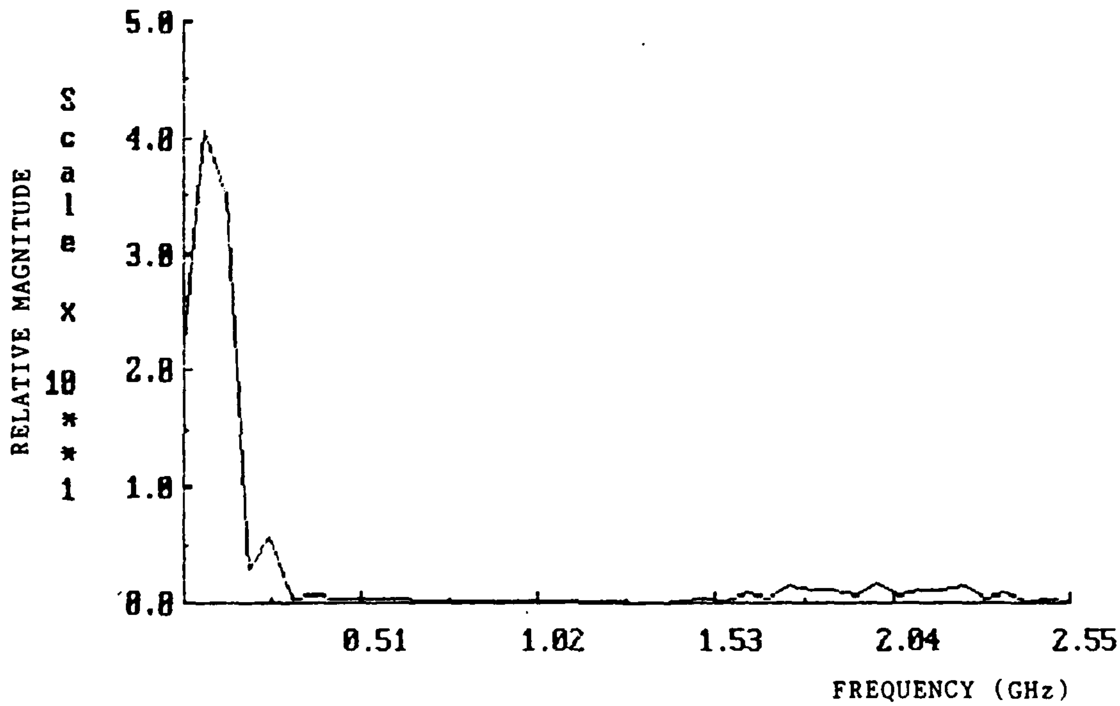


Figure 31. Frequency Response of Smoothed 6-Pole K-Pulse (B=1.01)

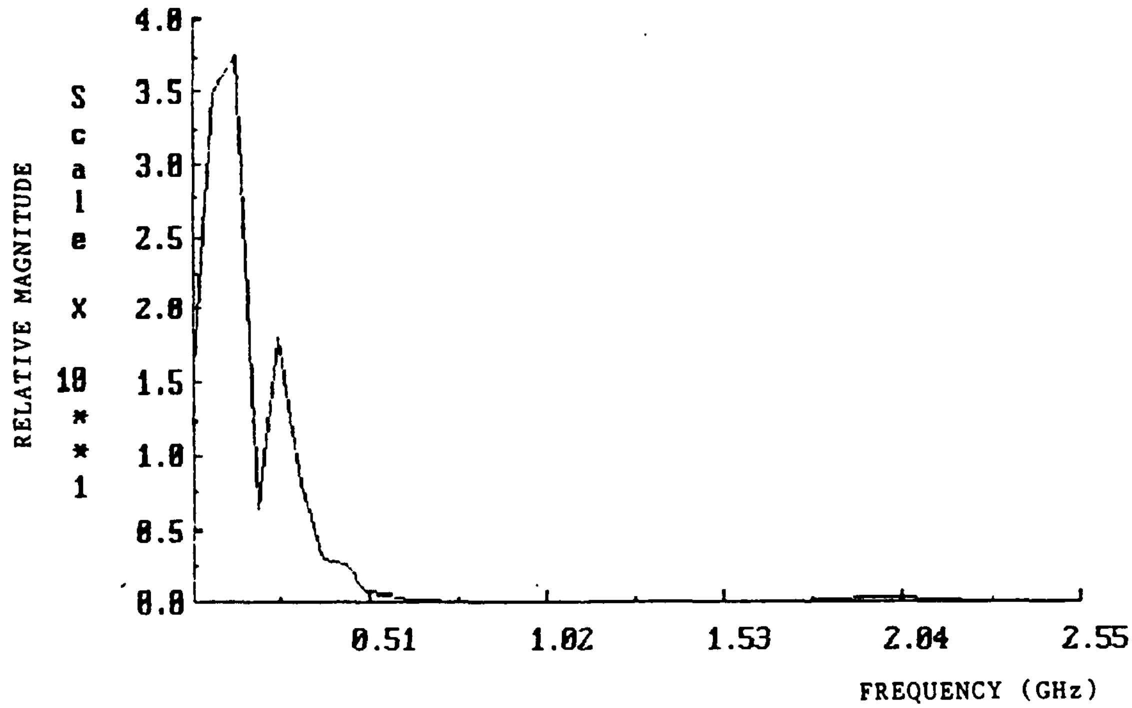


Figure 32. Frequency Response of Smoothed 6-Pole K-Pulse (B=2.0)

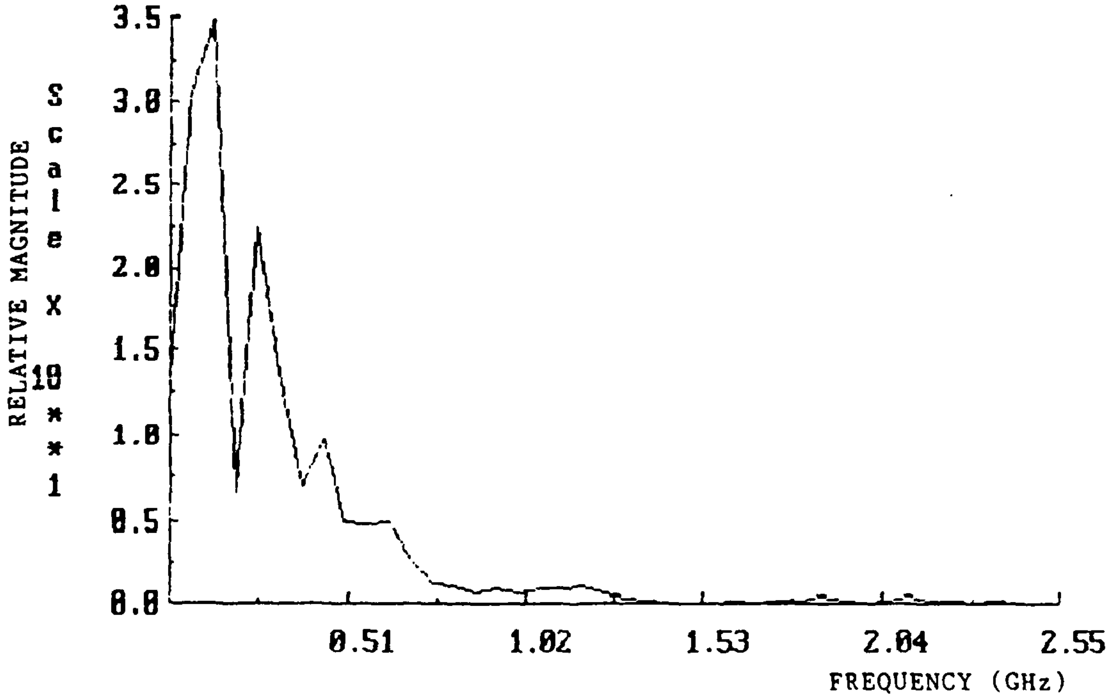


Figure 33. Frequency Response of Smoothed 6-Pole K-Pulse (B=3.0)

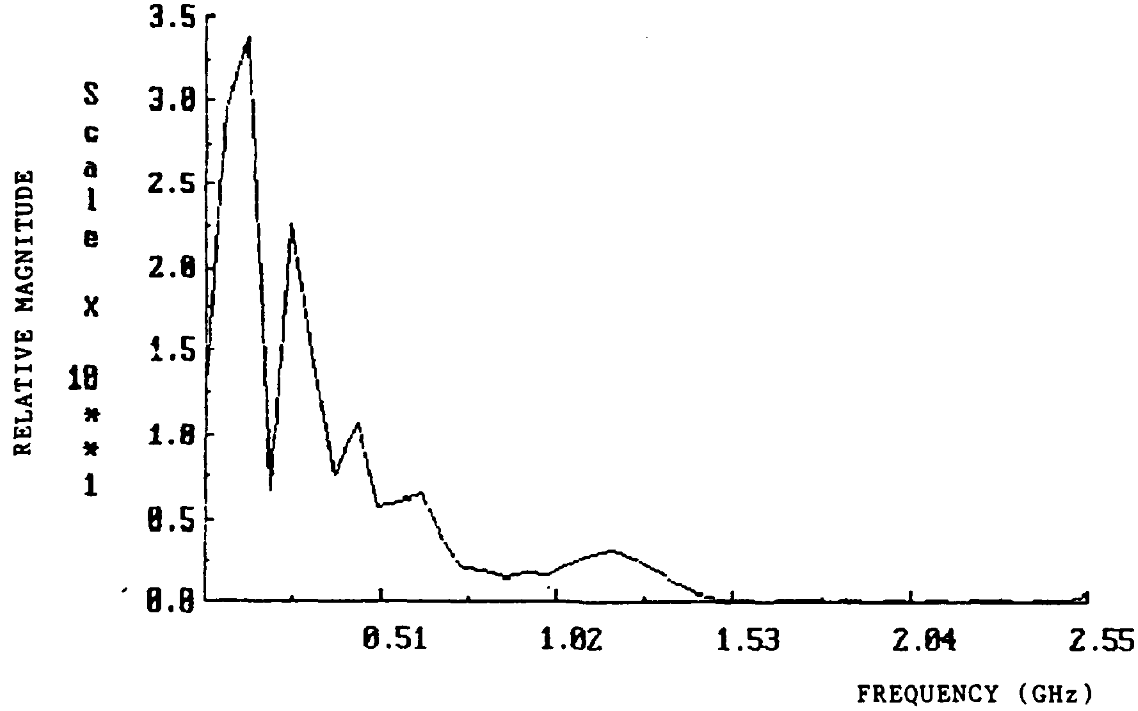


Figure 34. Frequency Response of Smoothed 6-Pole K-Pulse (B=3.2)

the spreading coefficient of 1.01 consistently provided the lowest energy output in the late-time for all noisy test signals. Results of the late-time energy outputs for K-Pulse preliminary testing are tabulated in Table 6.

TABLE 6. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR K-PULSES AND PRELIMINARY TEST WAVEFORMS

# of Poles	SNR	K-Pulses				
		Unsmoothed	B=1.01	B=2.0	B=3.0	B=3.2
1	∞	3.926819ε-10	4.985491ε-10	3.071268ε-10	9.837198ε-11	8.208115ε-11
	30dB	2.369440ε-04	1.304459ε-06	2.272184ε-05	2.590717ε-06	2.646095ε-06
	20dB	2.342825ε-03	1.288532ε-05	2.243572ε-05	2.559288ε-05	2.614233ε-05
	10dB	2.131517ε-02	1.176446ε-04	2.048043ε-04	2.336553ε-04	2.386780ε-04
2	∞	3.277037ε-10	1.053499ε-09	1.504356ε-09	9.078946ε-11	5.698292ε-11
	30dB	1.195194ε-04	5.975543ε-07	8.968465ε-07	1.465559ε-06	1.481735ε-06
	20dB	1.186214ε-03	5.951504ε-06	8.916999ε-06	1.453860ε-05	1.469953ε-05
	10dB	1.125299ε-02	5.565424ε-05	8.466547ε-05	1.379003ε-04	1.394275ε-04
4	∞	2.628599ε-10	1.258491ε-09	3.467784ε-09	1.59471ε-09	6.432455ε-10
	30dB	6.145333ε-05	2.935986ε-07	3.478439ε-07	5.425841ε-07	5.861888ε-07
	20dB	6.122306ε-04	2.900763ε-06	3.401900ε-06	5.378261ε-06	5.826654ε-06
	10dB	5.955821ε-03	2.816757ε-05	3.296361ε-05	5.226609ε-05	5.665313ε-05
6	∞	2.600232ε-10	1.0957ε-07	3.109389ε-07	3.592839ε-07	3.501574ε-07
	30dB	4.168548ε-05	3.220138ε-07	5.884405ε-07	6.768307ε-07	6.753122ε-07
	20dB	4.158295ε-04	2.086881ε-06	2.639911ε-06	3.086405ε-06	3.168090ε-06
	10dB	4.078638ε-03	1.090644ε-05	2.191860ε-05	2.574333ε-05	2.667117ε-05

An example of the cancellation obtained by the unsmoothed K-Pulse against the noiseless signals is displayed in Figure 35 for the 6-pole case. The large discontinuities present at the late-time transition are due to the fact that

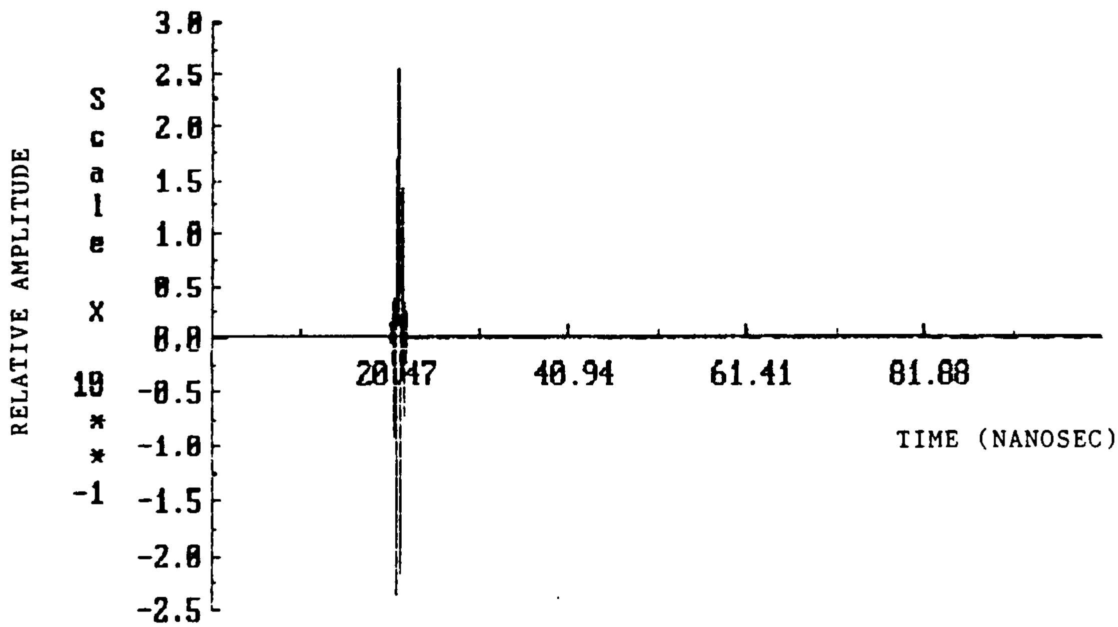


Figure 35. Convolution Output of Noiseless 6-Pole Test Signal and Unsmoothed K-Pulse

the entire K-Pulse is not operating on the late-time portion of the signal, and cancellation does not occur. This transition time is not part of the convolution output late-time energy calculation. Once the entire K-Pulse is operating on the signal, the cancellation begins. There is also an increase in the amplitude of the output waveform at the very end of the signal. This is due to the leading edge of the K-Pulse passing beyond the end of the signal, and the total cancellation effect is again lost. However, as the signal energy at the extreme end of the late-time is typically quite low, this is not considered to have a significant impact on the late-time energy calculation.

A plot of the frequency spectrum of the convolution output in the noiseless 6-pole case is displayed in Figure 36. When compared with Figure 4, the original spectrum of the noiseless 6-pole signal, the high degree of cancellation obtained is evident. This case is representative of the convolution output for the unsmoothed K-Pulses and other noiseless signals. Figure 37 is an illustration of the convolution output of the 4-pole signal with a SNR of 20 dB and the unsmoothed 4-pole K-Pulse. The uncancelled high frequency noise components are clearly visible in this illustration, and in the frequency domain representation, Figure 38. These plots are representative of the data obtained for the convolution outputs of the unsmoothed K-Pulses in a noisy environment.

Figure 39 displays the cancellation obtained when a smoothed 6-pole K-Pulse using a spreading coefficient of 1.01 is convolved with a test signal with a SNR of 10 dB. Figure 40 is the frequency domain representation of this

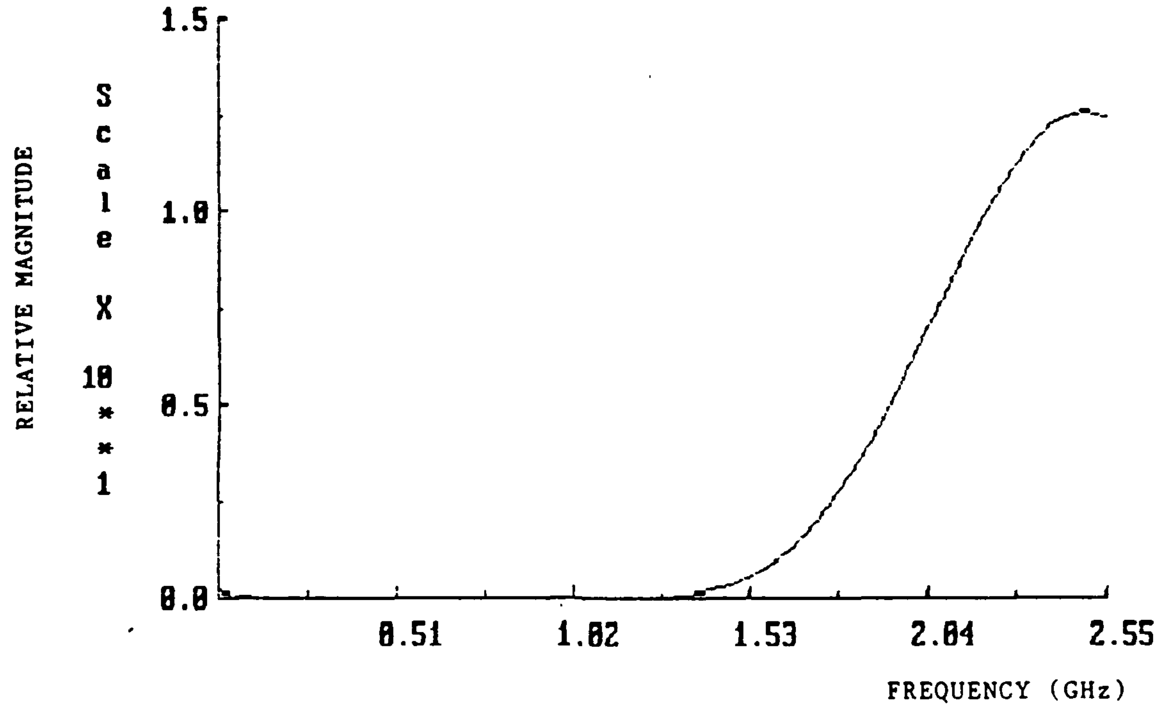


Figure 36. Frequency Spectrum of Convolution Output of Noiseless 6-Pole Test Signal and Unsmoothed K-Pulse

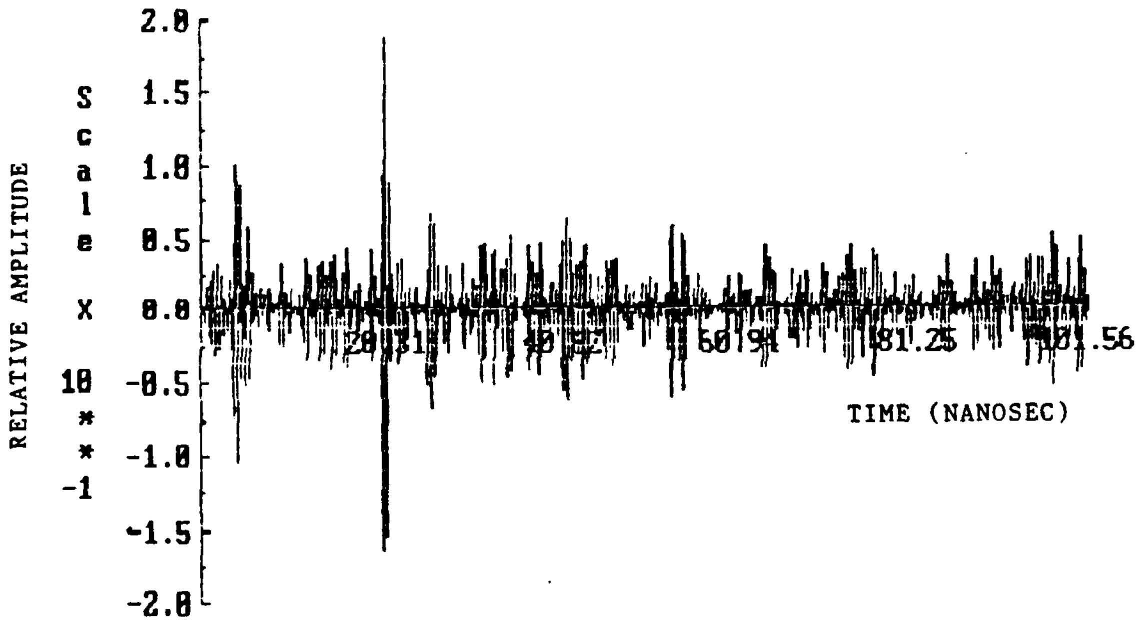


Figure 37. Convolution Output of 4-Pole Test Signal (SNR=20 dB) and Unsmoothed K-Pulse

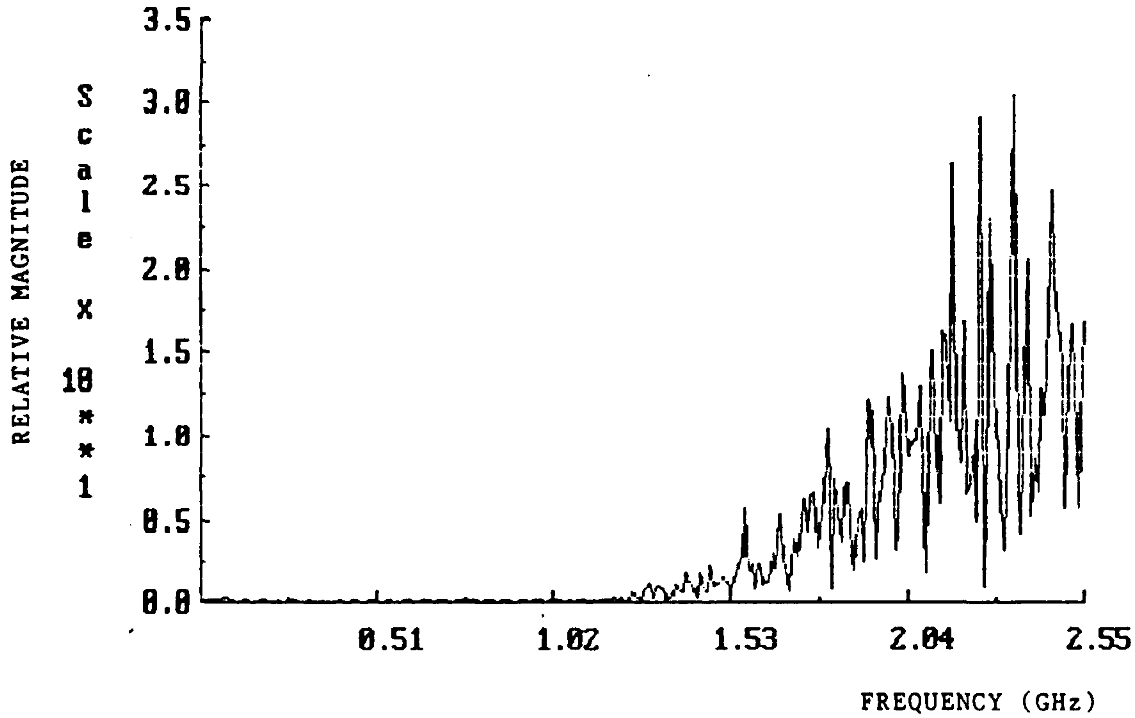


Figure 38. Frequency Spectrum of Convolution Output of 4-Pole Test Signal (SNR=20 dB) and Unsmoothed K-Pulse

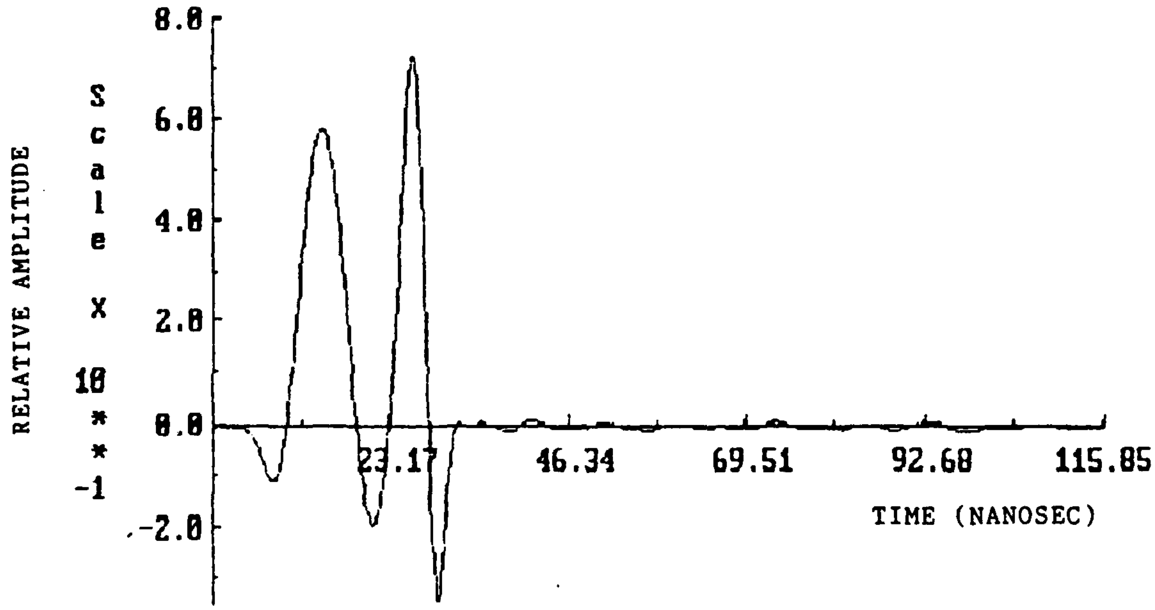


Figure 39. Convolution Output of 6-Pole Test Signal (SNR=10 dB) and Smoothed K-Pulse (B=1.01)

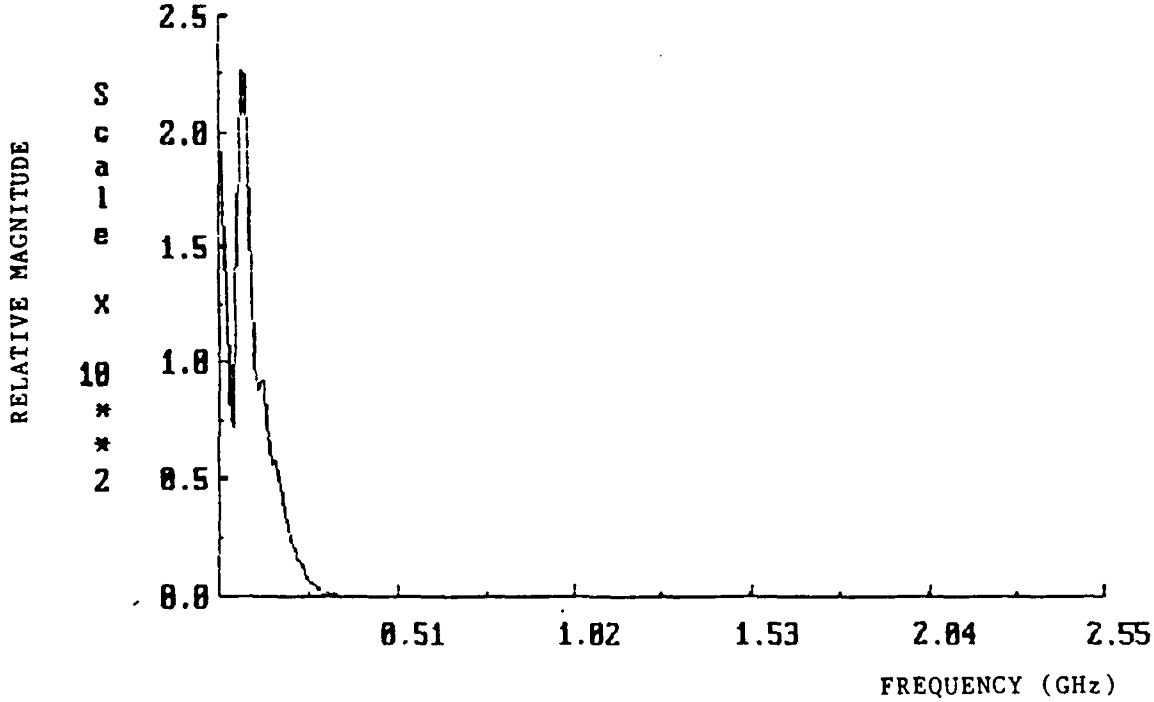


Figure 40. Frequency Spectrum of Convolution Output of 6-Pole Test Signal (SNR=10 dB) and Smoothed K-Pulse (B=1.01)

convolution output. Figure 41 illustrates the frequency components of the convolution output of the 6-pole test signal with a SNR of 10 dB and the K-Pulse using a spreading coefficient of 3.0.

3. Discrimination Effectiveness Testing

In discrimination effectiveness testing, K-Pulses were constructed from the pole data used to form the high-Q, medium-Q, and low-Q test signals (Tables 3-5). These poles were then varied five percent above and five percent below those values, to construct K-Pulses that would bracket each test signal and simulate a filter bank constructed for three targets with only slight variations in their late-time natural resonance responses. The two types of K-Pulses used in this test were the smoothed K-Pulse constructed using a spreading coefficient of 1.01, and the smoothed K-Pulse constructed using a spreading coefficient of 3.0.

The smoothed K-Pulse using the spreading coefficient of 1.01 was chosen for this test because it consistently had the lowest output energy in the late-time during preliminary testing. The smoothed K-Pulse using a spreading coefficient of 3.0 was chosen because of its deep nulls corresponding to the signal poles. The two K-Pulses were then compared to see which would provide the best target discrimination under varying noise conditions.

Figure 42 is the frequency spectrum of the smoothed high-Q K-Pulse using a spreading coefficient of 1.01. Figure 43 presents the same information for the smoothed high-Q K-Pulse using a spreading coefficient of 3.0. The other K-Pulses used in these tests are not displayed due to their similarity to these K-Pulses.

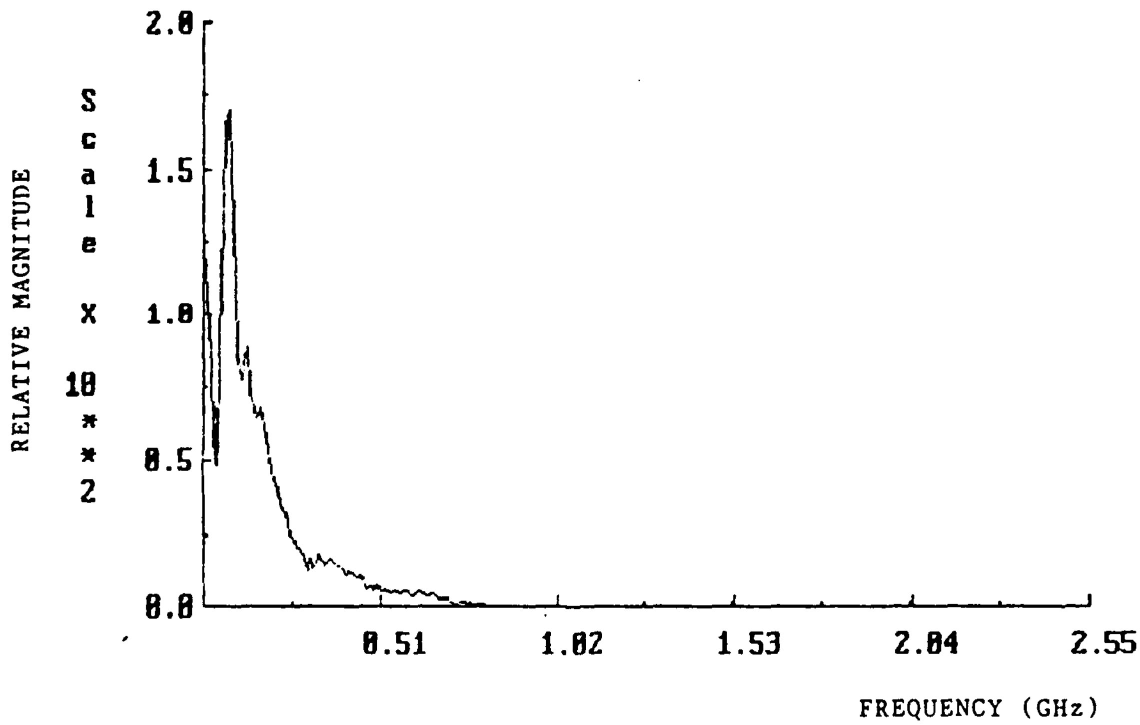


Figure 41. Frequency Spectrum of Convolution Output of 6-Pole Test Signal (SNR=10 dB) and Smoothed K-Pulse (B=3.0)

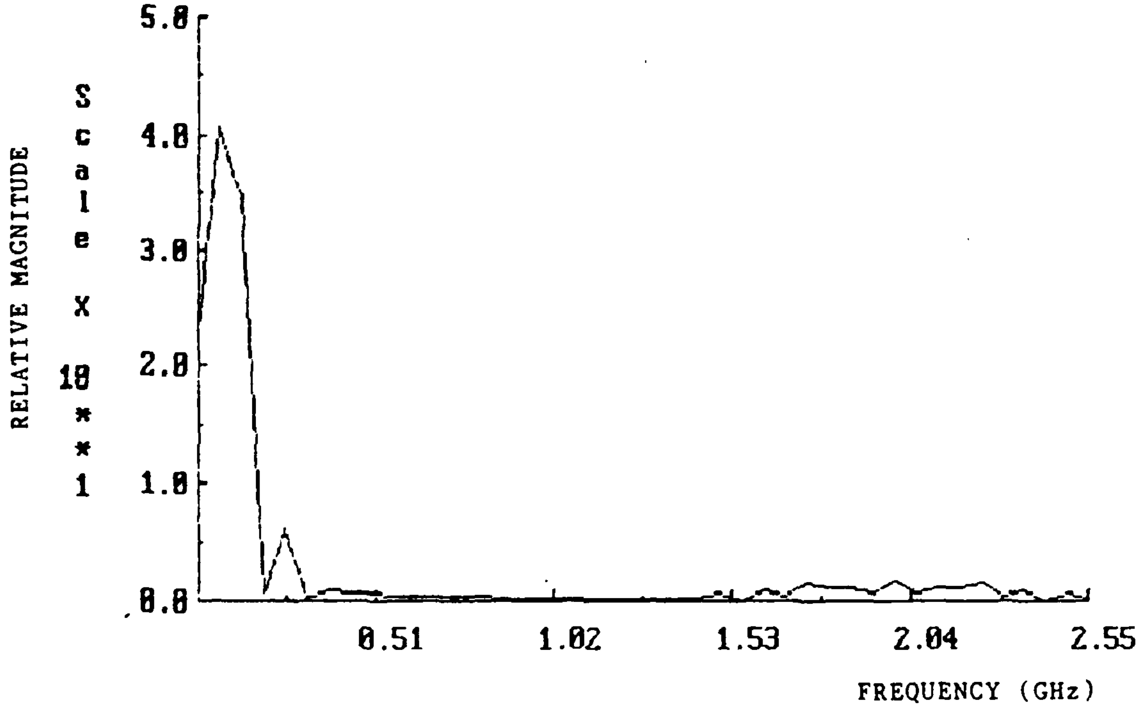


Figure 42. Frequency Response of Smoothed High-Q K-Pulse (B=1.01)

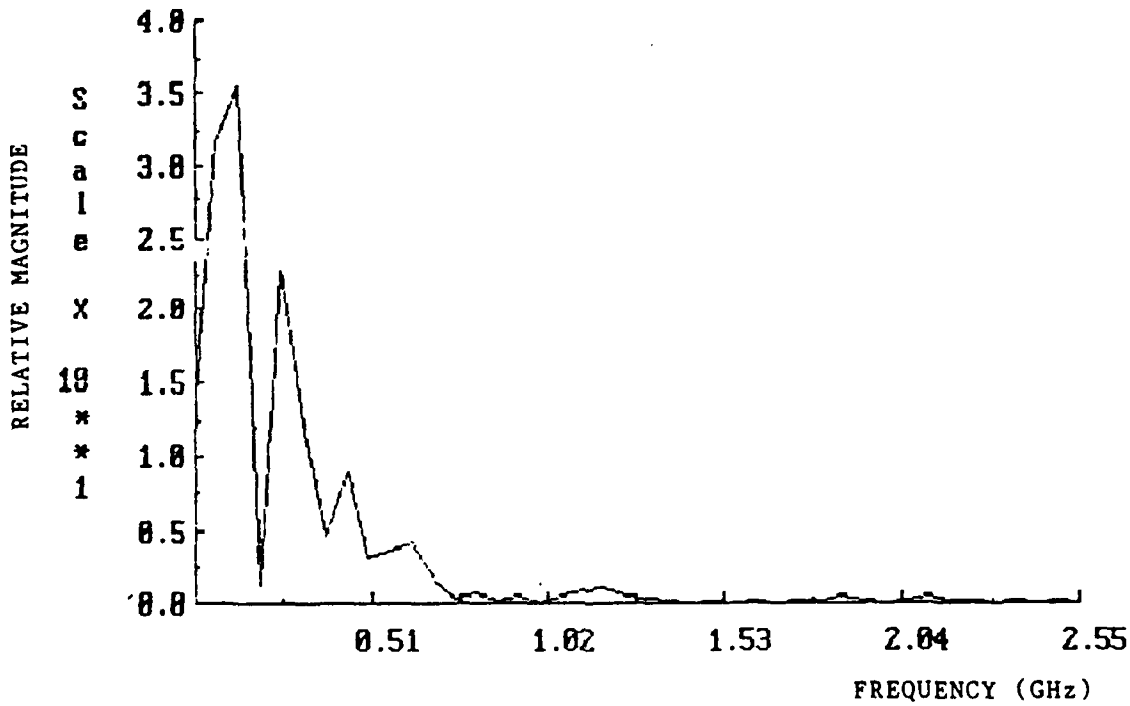


Figure 43. Frequency Response of Smoothed High-Q K-Pulse (B=3.0)

Both K-Pulses provide target discrimination for the high-Q and medium-Q test signals in environments with a SNR as low as 10 dB. Data for these test signals is provided in Tables 7a and 7b. A display of the convolution output for the 20 dB SNR medium-Q signal and the K-Pulse using a spreading coefficient of 3.0 is provided in Figure 44. The frequency spectrum is shown in Figure 45, and illustrates the virtually complete extinction of high frequency signal components. Displays for other conditions of noise and for the K-Pulse using a spreading coefficient of 1.01 are similar and are not included.

A review of data from Tables 7a and 7b indicates that the K-Pulse using a spreading coefficient of 3.0 has better discrimination characteristics than the K-Pulse using a spreading coefficient of 1.01. For the K-Pulse using the spreading coefficient of 3.0, in the medium-Q 20 dB SNR case, the ratio of the late-time energy outputs of the incorrect filters to the correct filter is at least 5:1. For the K-Pulse using the spreading coefficient of 1.01, only a minimum 2.7:1 ratio was obtained.

TABLE 7A. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR K-PULSE (B=1.01) AND HIGH-Q AND MEDIUM-Q TEST SIGNALS

Test Signal	K-Pulse (B=1.01)	SNR			
		∞	30dB	20dB	10dB
High-Q	5% above	8.764181ε-05	8.838877ε-05	9.123067ε-05	1.109269ε-04
	Exact	7.474967ε-06	7.832316ε-06	9.922313ε-06	2.807570ε-05
	5% below	5.805262ε-05	5.821318ε-05	5.981893ε-05	7.594170ε-05
Medium-Q	5% above	6.731178ε-06	7.281677ε-06	9.758017ε-06	2.856481ε-05
	Exact	6.007562ε-09	2.016770ε-07	1.911939ε-06	1.849535ε-05
	5% below	4.022191ε-06	3.990396ε-06	5.212148ε-06	2.030188ε-05

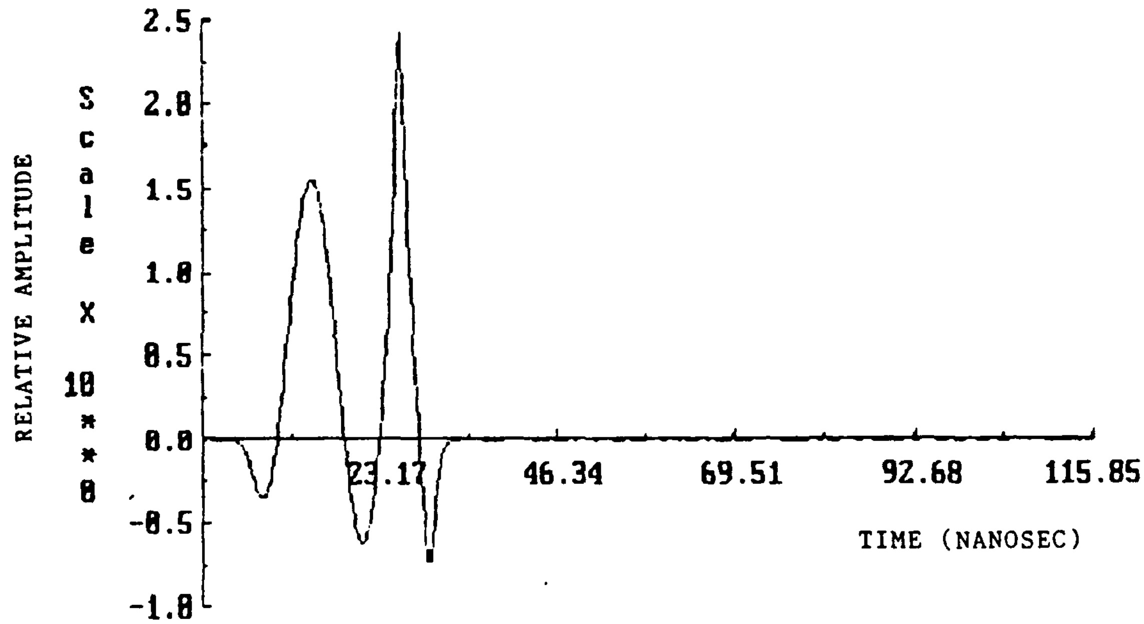


Figure 44. Convolution Output of Medium-Q Test Signal (SNR=20 dB) and Exact K-Pulse (B=3.0)

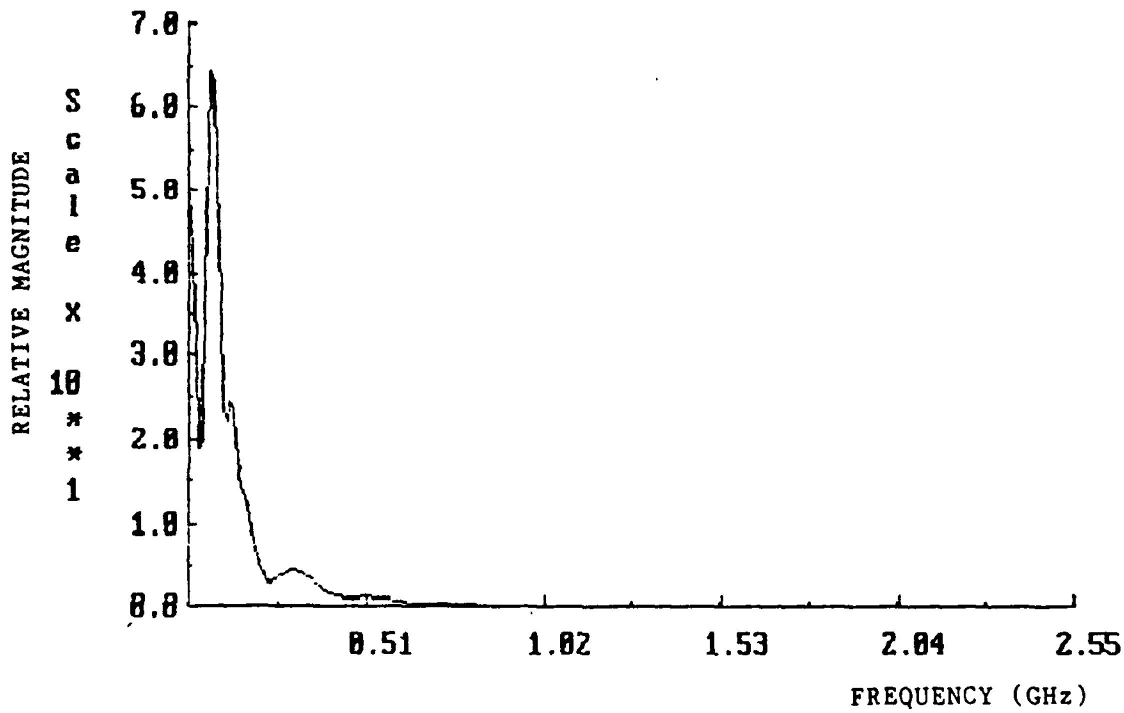


Figure 45. Frequency Spectrum of Convolution Output of Medium-Q Test Signal (SNR=20 dB) and Exact K-Pulse (B=3.0)

TABLE 7B. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR K-PULSE (B=3.0) AND HIGH-Q AND MEDIUM-Q TEST SIGNALS

Test Signal	K-Pulse (B=3.0)	SNR			
		∞	30dB	20dB	10dB
High-Q	5% above	2.681329ε-04	2.694126ε-04	2.735891ε-04	2.990642ε-04
	Exact	1.248741ε-05	1.291247ε-05	1.554313ε-05	3.886461ε-05
	5% below	2.719982ε-04	2.719309ε-04	2.731629ε-04	2.891731ε-04
Medium-Q	5% above	1.272588ε-05	1.341368ε-05	1.664759ε-05	4.199747ε-05
	Exact	1.670336ε-06	2.897475ε-07	2.643178ε-06	2.560697ε-05
	5% below	1.176944ε-05	1.172304ε-05	1.336586ε-05	3.372494ε-05

For the K-Pulse using the spreading coefficient of 3.0 in the medium-Q 10 dB SNR case, the ratio of the late-time energy of the incorrect filters to the correct filter was at least 1.3:1. For the K-Pulse using the spreading coefficient of 1.01, a ratio of only 1.1:1 was obtained.

For the low-Q signal, correct identifications were made by both K-Pulses in the noiseless case. Once only 30dB SNR noise was added to the signal, however, neither K-Pulse could properly discriminate against signals with a five percent variation in late-time characteristics from the correct signal. This is considered to be due to the lack of available late-time energy in the low-Q signal. Data for the low-Q signal is compiled in Table 8.

TABLE 8. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR K-PULSES AND LOW-Q TEST SIGNAL

K-Pulse		SNR			
		∞	30dB	20dB	10dB
	5% above	6.132034ε-09	1.870701ε-07	1.798446ε-06	1.749190ε-05
B=1.01	Exact	1.933096ε-12	1.791509ε-07	1.783530ε-06	1.742781ε-05
	5% below	1.987890ε-09	1.764799ε-07	1.750056ε-06	1.711700ε-05
	5% above	1.879068ε-09	3.465386ε-07	3.412570ε-06	3.326598ε-05
B=3.0	Exact	7.515653ε-13	3.395724ε-07	3.380869ε-06	3.303704ε-05
	5% below	1.240274ε-09	3.295206ε-07	3.297233ε-06	3.229405ε-05

C. RESULTS OF E-PULSE TESTING

1. Preliminary Testing

A Microsoft FORTRAN program implementing the matrix approach using a forcing function for creating E-Pulses was developed and the program listing included in Appendix F. Preliminary testing of the program was conducted using the 6-pole, 4-pole, 2-pole and 1-pole signals described in Section IIIA2 above. E-Pulses were created using rectangular pulses as basis functions, with widths that matched ($T=\Delta t$), doubled ($T=2\Delta t$), tripled ($T=3\Delta t$), and quadrupled ($T=4\Delta t$) the sampling period of the test signal. These E-Pulses were then convolved with the test signals to determine the width that would provide the least output energy in the late-time. The program listing for the FORTRAN implementation of this convolution is included in Appendix G.

The 6-pole E-Pulse created using the pole data found in Table 1, as well as pulse basis functions of the same width as the sampling period of the test signal, is displayed in Figure 46. The frequency spectrum is plotted in Figure 47. It is observed that the magnitude of the E-Pulse increases monotonically with frequency, and will thus amplify any high frequency noise components present in a signal. For comparison purposes, the 6-pole E-pulse with basis functions twice the width of the sampling period of the test signal is plotted in Figure 48, with its frequency spectrum displayed in Figure 49. It is observed that the magnitude of this E-Pulse decreases with frequency.

The 6-pole E-Pulse with a basis function width of triple the sampling period of the test signal is plotted in Figure 50, and its frequency spectrum is

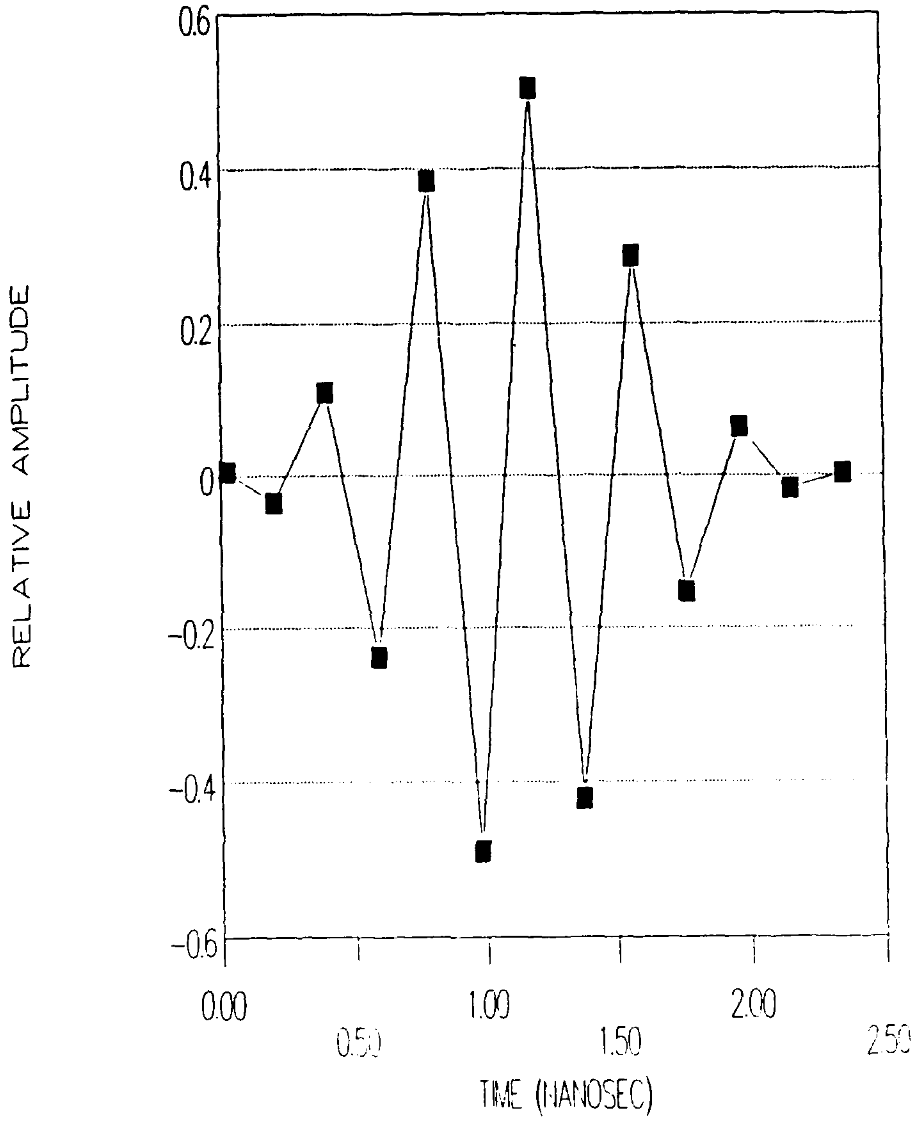


Figure 46. Time Domain Representation of 6-Pole E-Pulse (T=Δt)

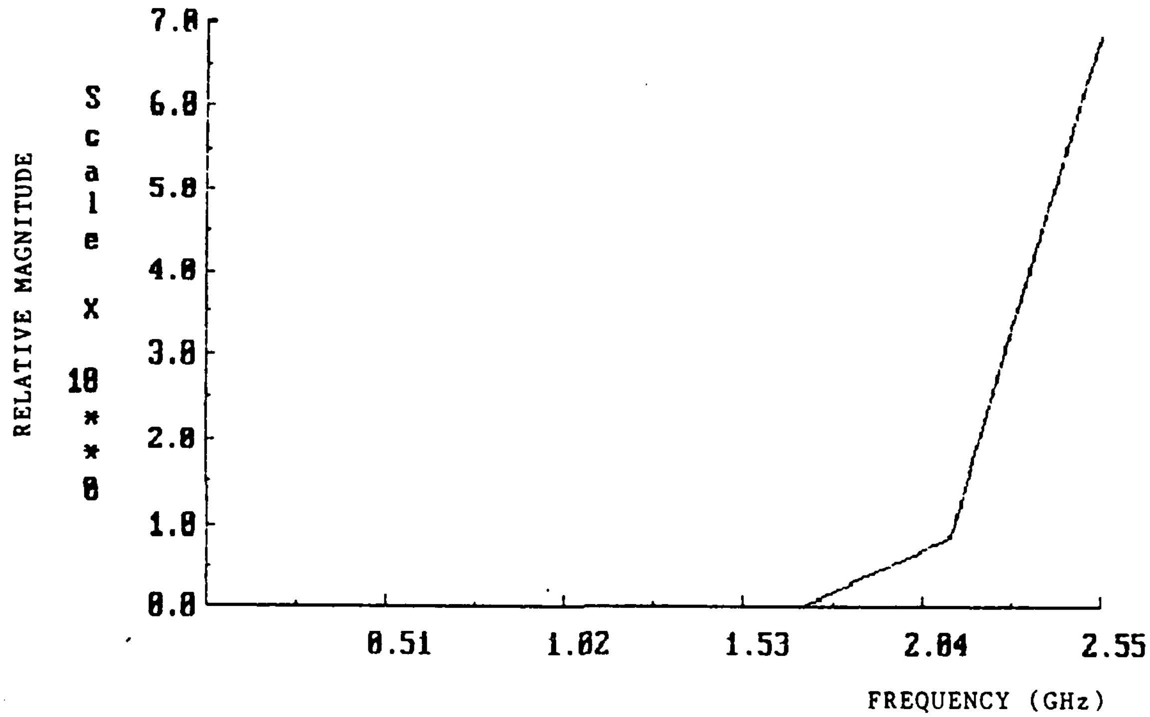


Figure 47. Frequency Response of 6-Pole E-Pulse (T=Δt)

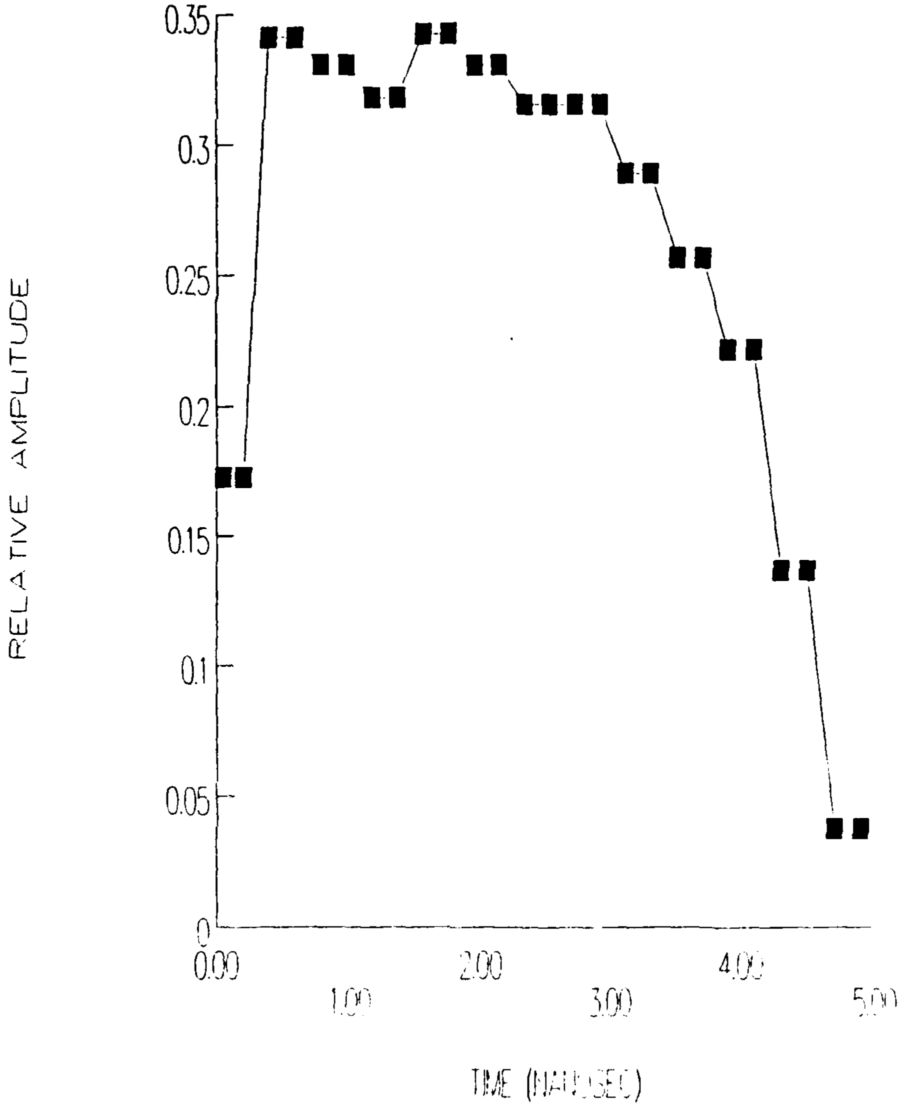


Figure 48. Time Domain Representation of 6-Pole E-Pulse (T=2Δt)

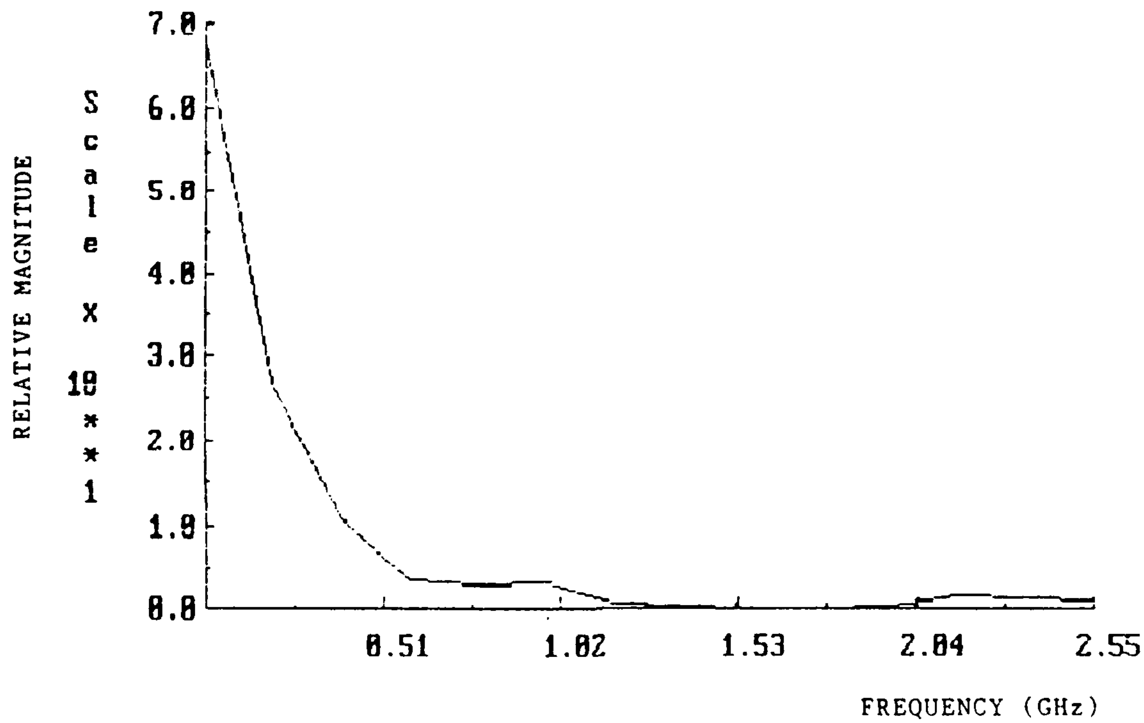
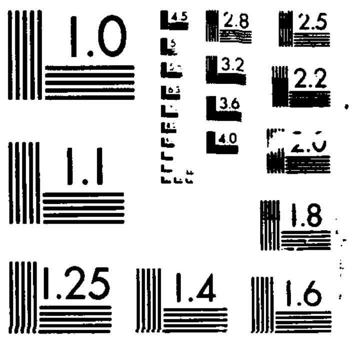


Figure 49. Frequency Response of 6-Pole E-Pulse (T=2Δt)

AD-A220 172 A COMPARISON OF THE K-PULSE AND E-PULSE TECHNIQUES FOR ASPECT INDEPENDENT RADAR TARGET IDENTIFICATION(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M S SIMON SEP 89 2/2

UNCLASSIFIED F/G 17/9 NL

END
FILMED
DTIC


1.0
4.5
2.8
2.5
3.2
2.2
3.6
4.0
1.1
1.8
1.25
1.4
1.6

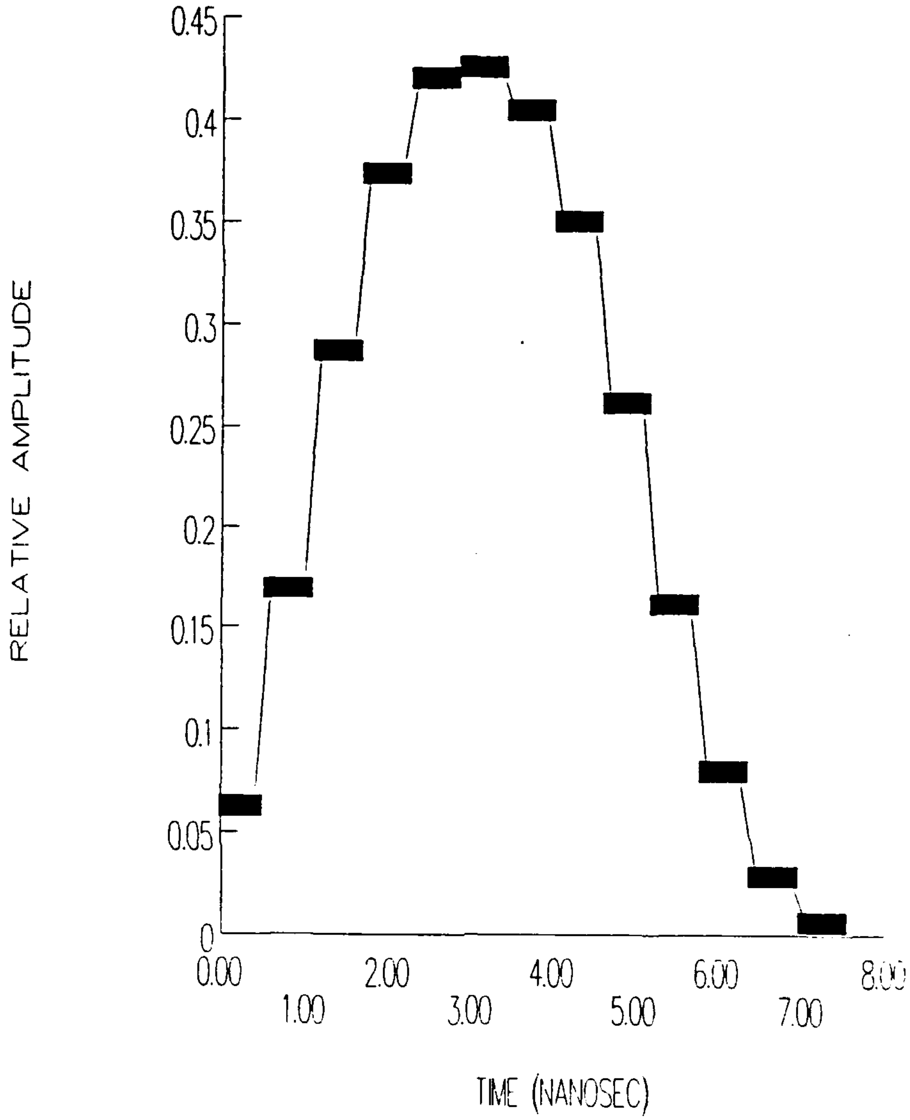


Figure 50. Time Domain Representation of 6-Pole E-Pulse (T=3Δt)

shown in Figure 51. The magnitude of this E-Pulse also decreases with frequency. The spectra of two representative four-pole E-Pulses are displayed in Figures 52 and 53.

Results indicate that the E-Pulse constructed using basis functions having widths identical to the sampling period of the test signal produce the lowest energy output in the late-time for all cases, with the exception of the 6-pole noisy signals. This is the expected outcome, given the frequency spectra of the E-Pulses and the test signals. In the 1-pole case, use of any E-Pulse other than the matched case will not reduce the noise components in the late-time, and will not completely cancel the late-time natural resonance itself. The 2-pole and 4-pole cases are similar to the 1-pole case. In the 6-pole case, the matched E-Pulse is ineffective in minimizing the late-time energy of the signal in a noisy environment because of the amplification of the high frequency noise components. Results of the late-time energy outputs for the 1-pole, 2-pole, and 4-pole cases are provided in Table 9.

A graphical display of the cancellation obtained by the matched sampling period E-Pulses is provided in Figure 54 for the noiseless 4-pole case. As demonstrated in the case of the K-Pulse, the large discontinuities at the late time transition are due to the fact that the entire E-Pulse is not operating on the late-time portion of the signal, and cancellation does not occur. Once the entire E-Pulse is operating on the signal, the cancellation begins. There is also an increase in the amplitude of the output waveform at the very end of the signal, as previously seen in the K-Pulse. This is due to the leading edge of the E-Pulse passing over the end of the signal, and the total cancellation effect is again lost.

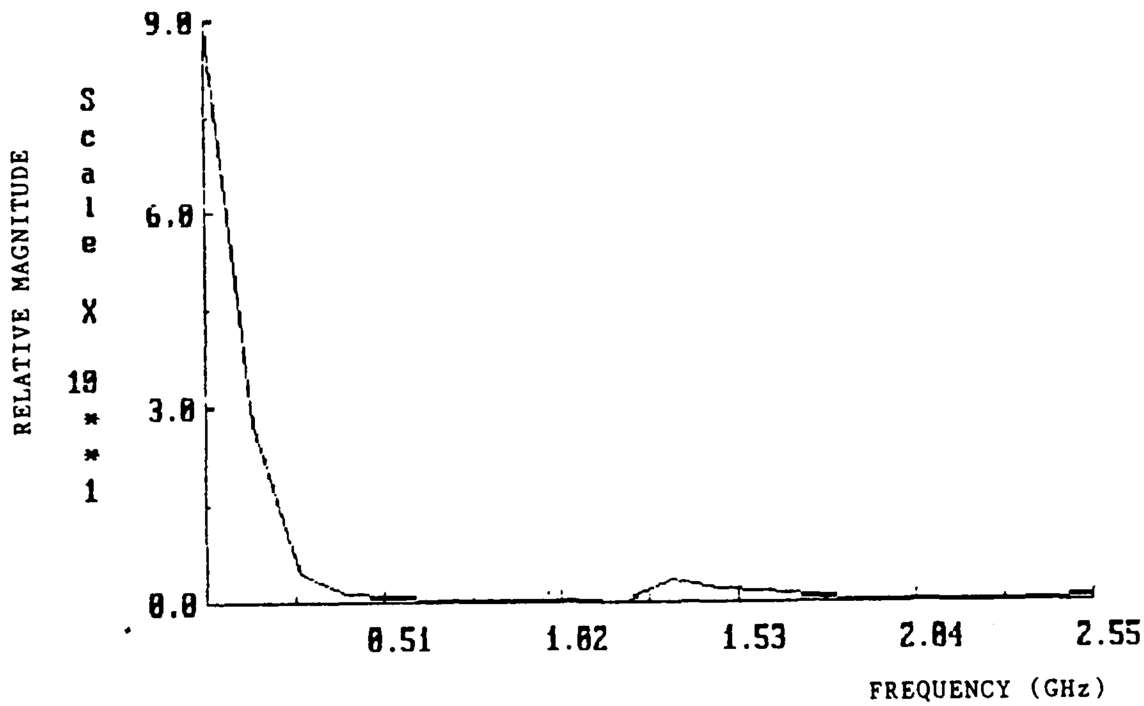


Figure 51. Frequency Response of 6-Pole E-Pulse (T=3Δt)

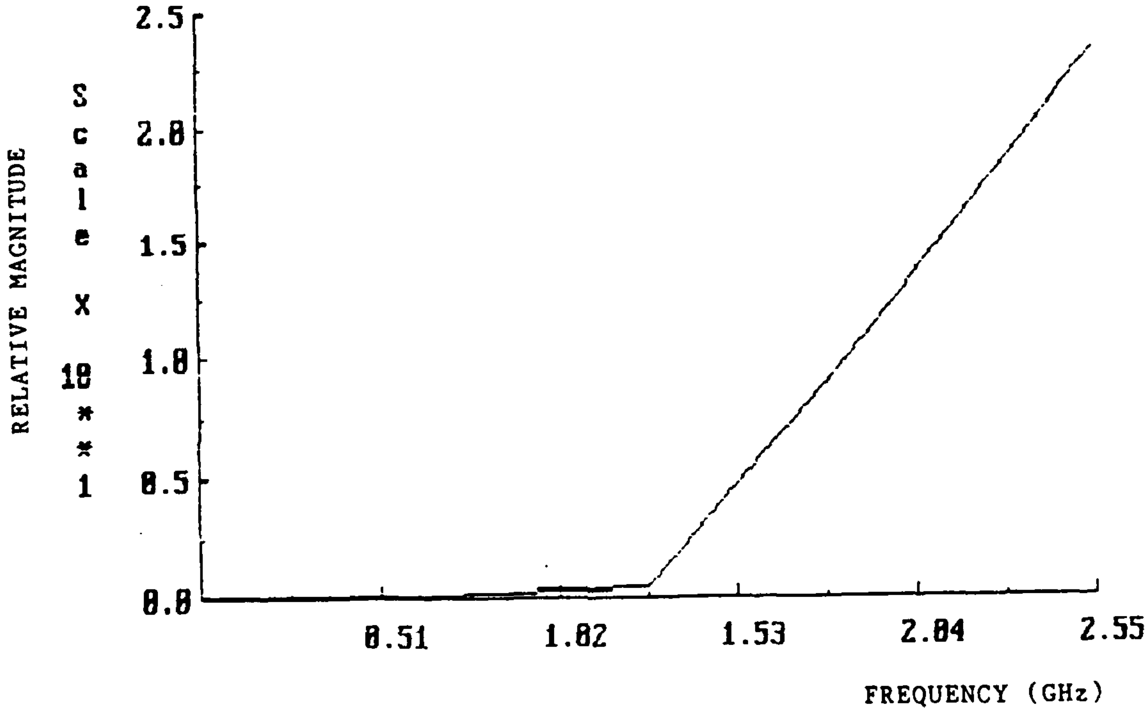


Figure 52. Frequency Response of 4-Pole E-Pulse (T=Δt)

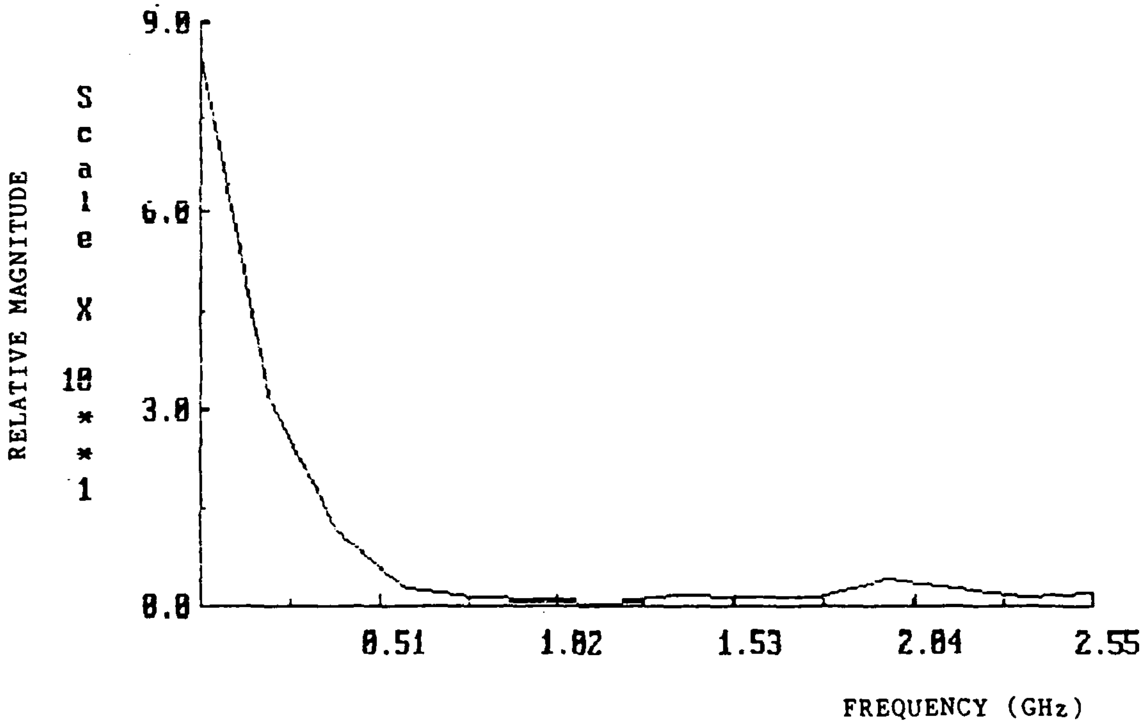


Figure 53. Frequency Response of 4-Pole E-Pulse (T=3Δt)

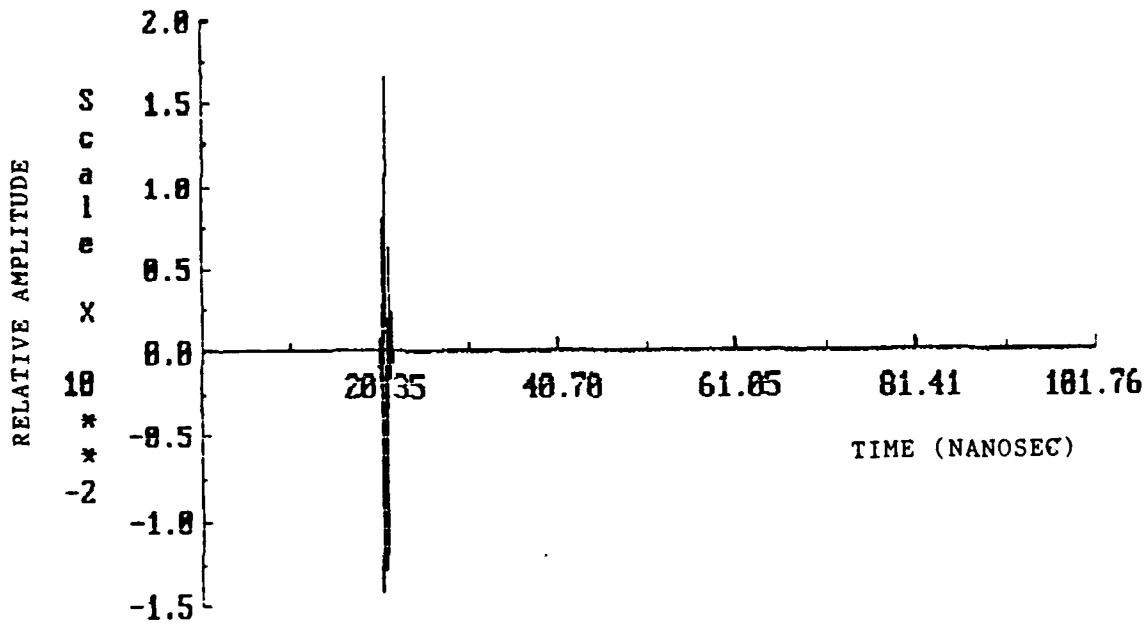


Figure 54. Convolution Output of Noiseless 4-Pole Test Signal and 4-Pole E-Pulse (T=Δt)

TABLE 9. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR E-PULSES AND PRELIMINARY TEST WAVEFORMS

# of Poles	SNR	E-Pulse			
		T=ΔT	T=2ΔT	T=3ΔT	T=4ΔT
1	∞	3.926861ε-12	2.400060ε-11	1.005212ε-10	3.352055ε-10
	30dB	2.369441ε-06	5.281731ε-06	8.693177ε-06	9.978448ε-06
	20dB	2.342825ε-05	5.221239ε-05	9.592171ε-05	9.858569ε-05
	10dB	2.13957ε-04	4.767831ε-04	7.845649ε-04	9.000952ε-04
2	∞	3.278894ε-12	1.563016ε-11	8.499395ε-11	7.596687ε-10
	30dB	1.195194ε-06	2.567914ε-06	4.534117ε-06	2.918845ε-06
	20dB	1.186213ε-05	2.547789ε-05	4.498048ε-05	2.892009ε-05
	10dB	1.125299ε-04	2.416694ε-04	4.266445ε-04	2.742154ε-04
4	∞	2.628933ε-12	2.211800ε-11	5.249828ε-10	5.539259ε-10
	30dB	6.145331ε-07	1.2344ε-06	8.185941ε-07	9.670603ε-07
	20dB	6.122306ε-06	1.229897ε-05	8.129678ε-06	9.600685ε-06
	10dB	5.955825ε-05	1.195197ε-04	7.90119ε-05	9.329743ε-05

A plot of the frequency spectrum of the convolution output in the 4-pole noiseless case, displayed in Figure 55, indicates the degree of cancellation of the signal poles when compared with Figure 8, the frequency spectrum of the original 4-pole test signal. It is clear that the only part of the original spectrum not cancelled is the low frequency portion associated with the early-time response, including any DC component.

Figure 56 displays the cancellation obtained for the 2-pole test signal with a SNR of 10 dB convolved with the E-Pulse with basis functions of width equal to the test signal sampling period. It is observed that the signal is extinguished but the high frequency noise components have been amplified.

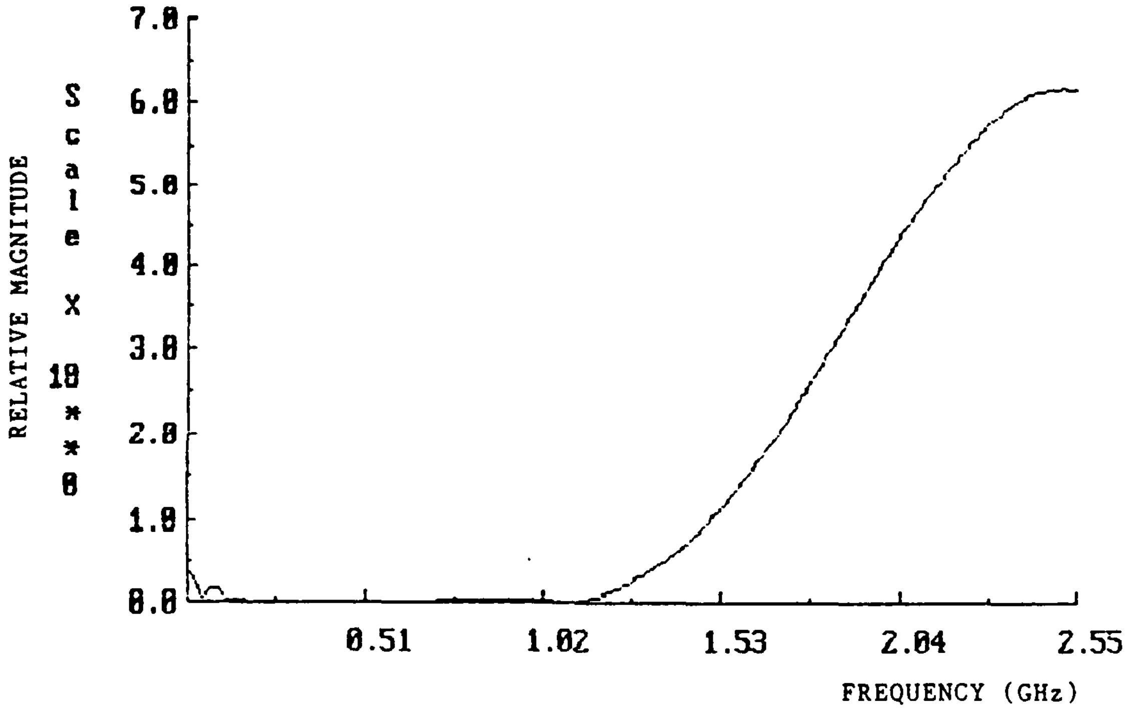


Figure 55. Frequency Spectrum of Convolution Output of 4-Pole Noiseless Test Signal and 4-Pole E-Pulse (T=Δt)

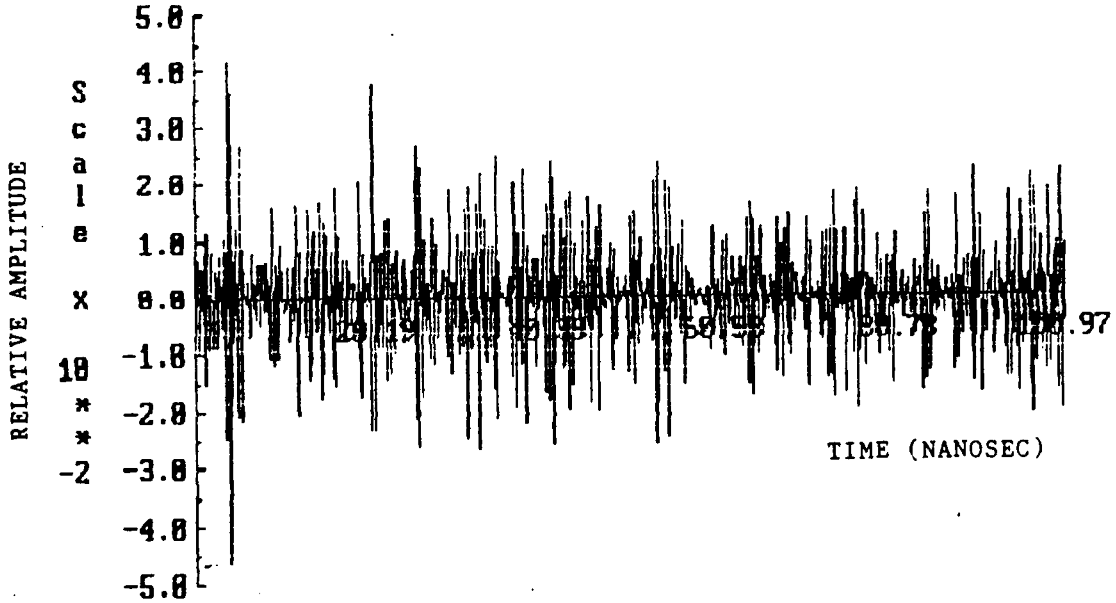


Figure 56. Convolution Output of 2-Pole Test Signal (SNR=10 dB) and 2-Pole E-Pulse (T=Δt)

The frequency domain representation of this convolution output, Figure 57, confirms this observation. These plots are representative of the data obtained for the 1-pole case and other conditions of noise in the 2-pole and 4-pole cases.

In the six-pole case, the E-Pulse that used basis functions twice as wide as that of the test signal provided the lowest late-time energy output when convolved with the noisy signals. The plot of its frequency spectrum, given in Figure 49, indicates that this E-pulse provides the optimum cancellation of high frequency components, while still providing a high degree of cancellation of the late-time natural resonances and other lower band frequency components. Table 10 is a compilation of results obtained for the 6-pole test waveforms. Figure 58 displays the convolution output of the 6-pole test signal with 10 dB SNR and the E-Pulse using basis functions twice as wide as the sampling period of the signal, and Figure 59 illustrates the frequency spectrum. The virtually complete cancellation of the high frequency components is clear.

TABLE 10. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR E-PULSES AND 6-POLE PRELIMINARY TEST SIGNAL

SNR	E-Pulse			
	T=ΔT	T=2ΔT	T=3ΔT	T=4ΔT
∞	2.599018ε-12	3.662053ε-10	4.310820ε-10	7.723869ε-10
30dB	4.168548ε-07	3.751241ε-07	4.739370ε-07	1.293498ε-06
20dB	4.158295ε-06	3.726649ε-06	4.705832ε-06	1.286415ε-05
10dB	4.078640ε-05	3.651180ε-05	4.609681ε-05	1.260722ε-04

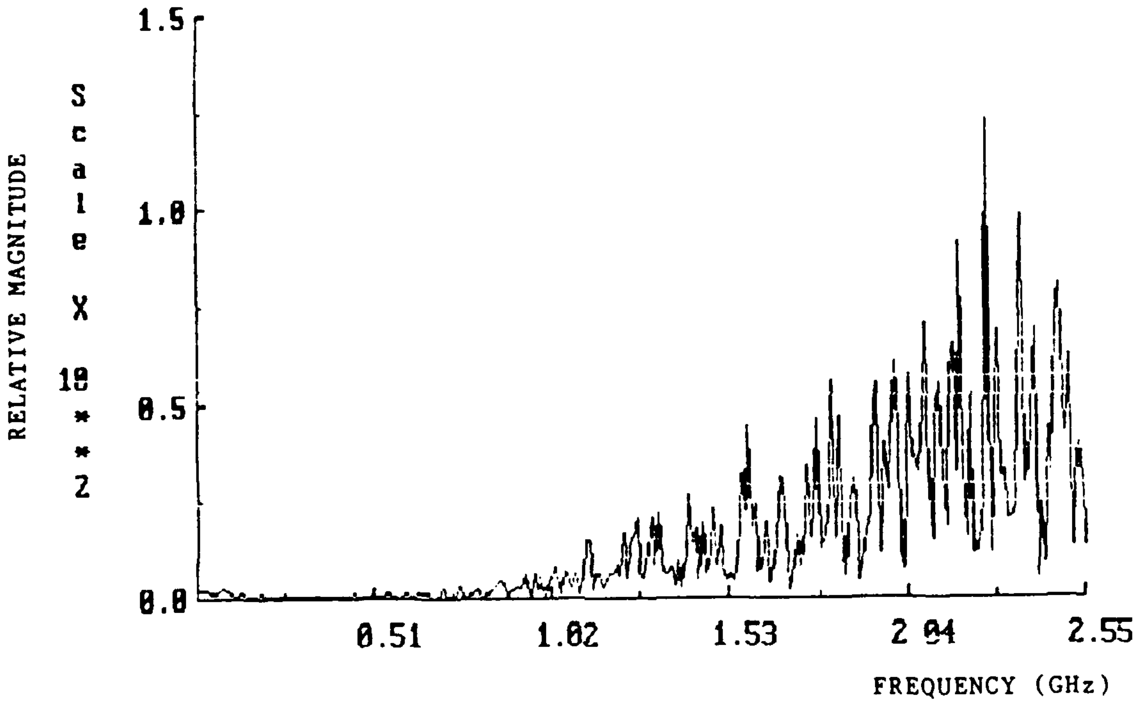


Figure 57. Frequency Spectrum of Convolution Output of 2-Pole Test Signal (SNR=10 dB) and 2-Pole E-Pulse (T=Δt)

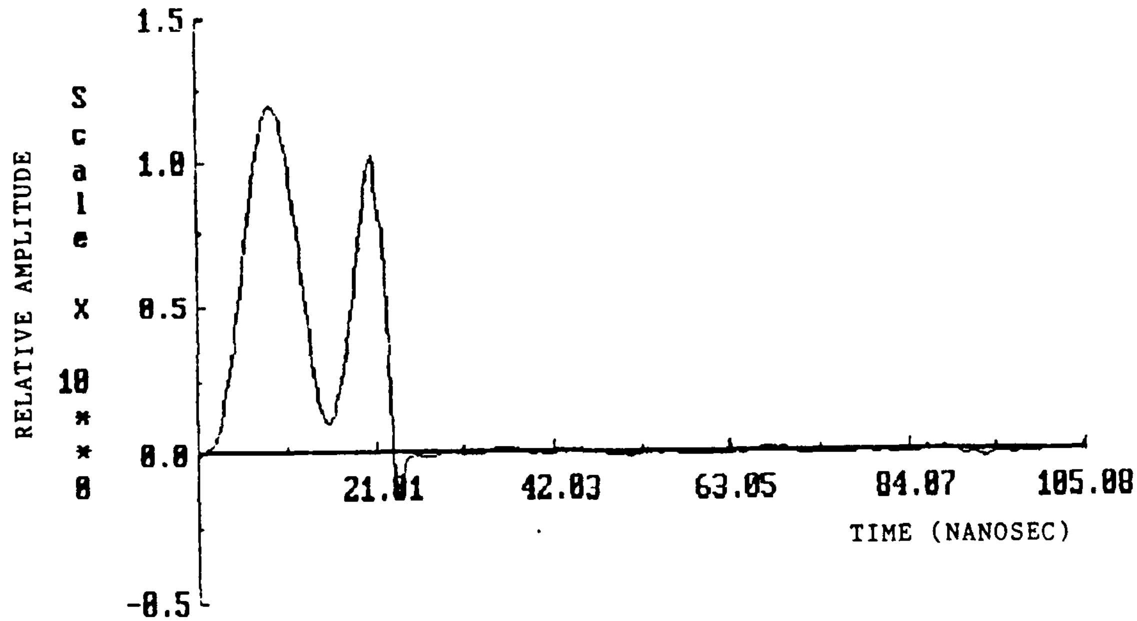


Figure 58. Convolution Output of 6-Pole Test Signal (SNR=10 dB) and 6-Pole E-Pulse (T=2Δt)

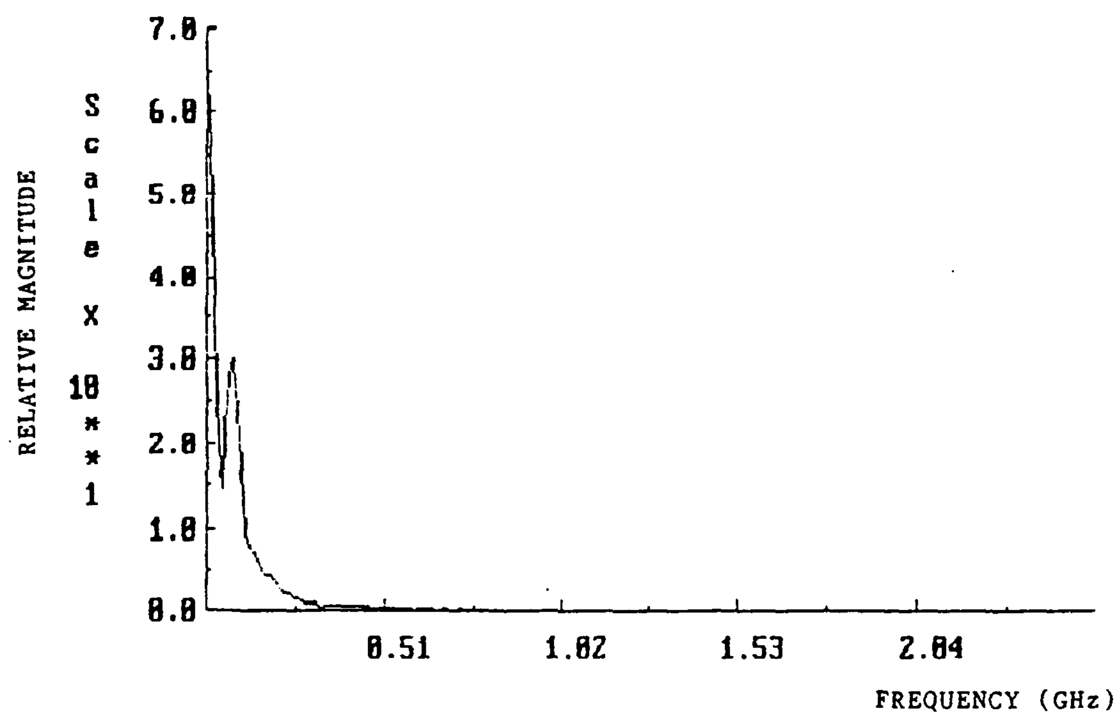


Figure 59. Frequency Spectrum of Convolution Output of 6-Pole Test Signal (SNR=10 dB) and 6-Pole E-Pulse (T=2Δt)

2. Discrimination Effectiveness Testing

In discrimination effectiveness testing, E-Pulses were constructed from the same pole data used to form the high-Q, medium-Q, and low-Q test signals (Tables 3-5), duplicating the conditions created for K-Pulse discrimination effectiveness testing. These poles were then varied five percent above and five percent below those values, to construct E-Pulses that would bracket each test signal and simulate a filter bank constructed for three targets with only slight variations in their late-time natural resonance characteristics. Two classes of E-Pulses were constructed; the first composed of pulse basis functions with widths to match the sampling period of the test signals; the second used pulse basis functions with widths of twice the sampling period of the test signal.

The exact E-Pulses constructed for the high-Q and medium-Q test signals are not displayed due to their virtual identity with the 6 pole E-Pulse used in preliminary testing. The time domain representations and frequency spectra for the narrow and wide, low-Q E-Pulses are plotted in Figures 60 through 63.

For the high-Q signal under noiseless and 30 dB SNR conditions, and for the noisy low-Q signals, the exact E-Pulse having basis functions whose width matched the sampling period of the test signal provided the lowest output energy in the late time. In all other cases, no discrimination was made, and incorrect identification resulted. Table 11 presents the convolution output data for the narrow E-Pulse and the high-Q, medium-Q, and low-Q test signals.

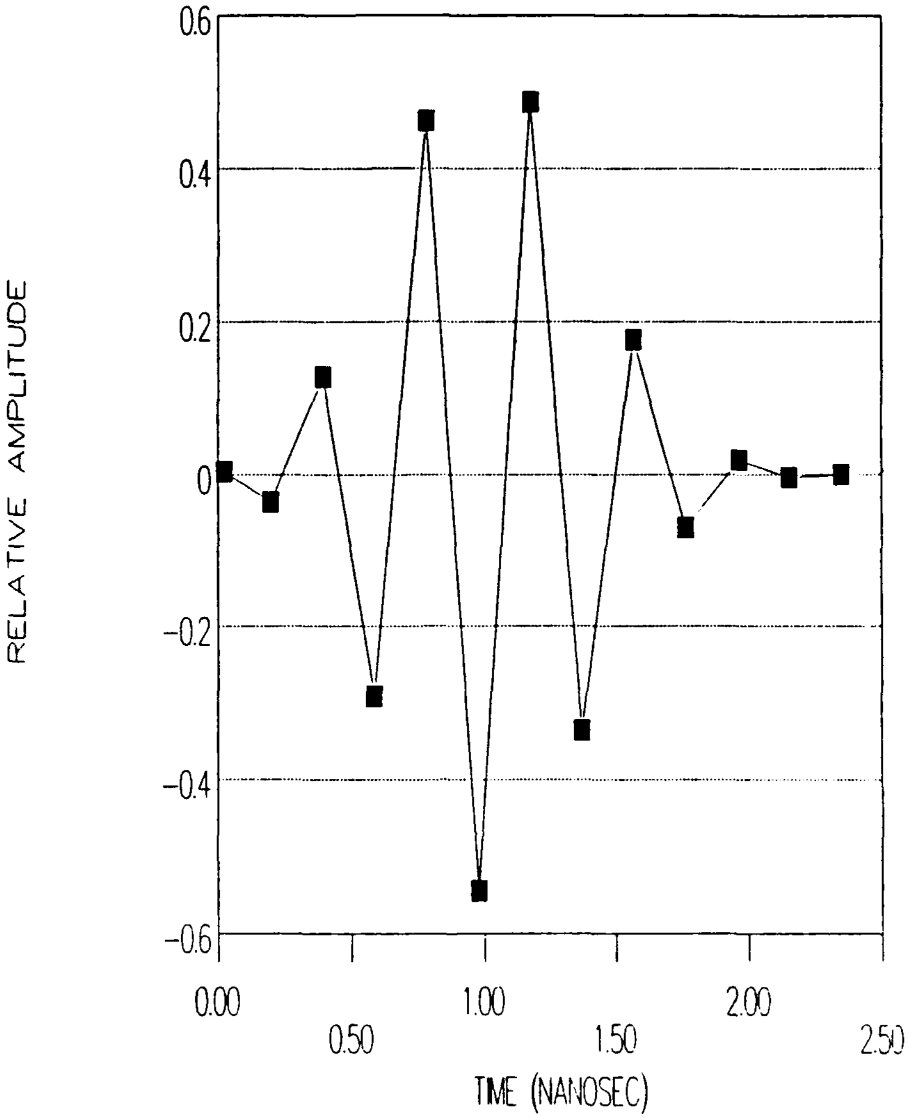


Figure 60. Time Domain Representation of Narrow Low-Q Exact E-Pulse

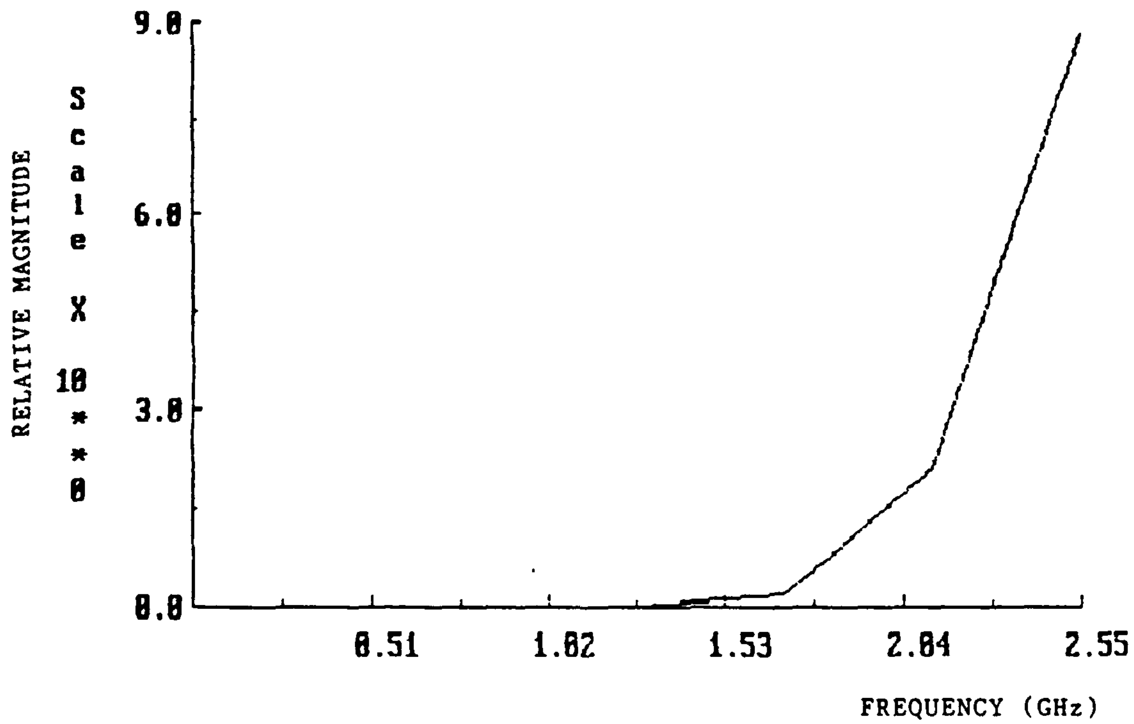


Figure 61. Frequency Response of Narrow Low-Q Exact E-Pulse

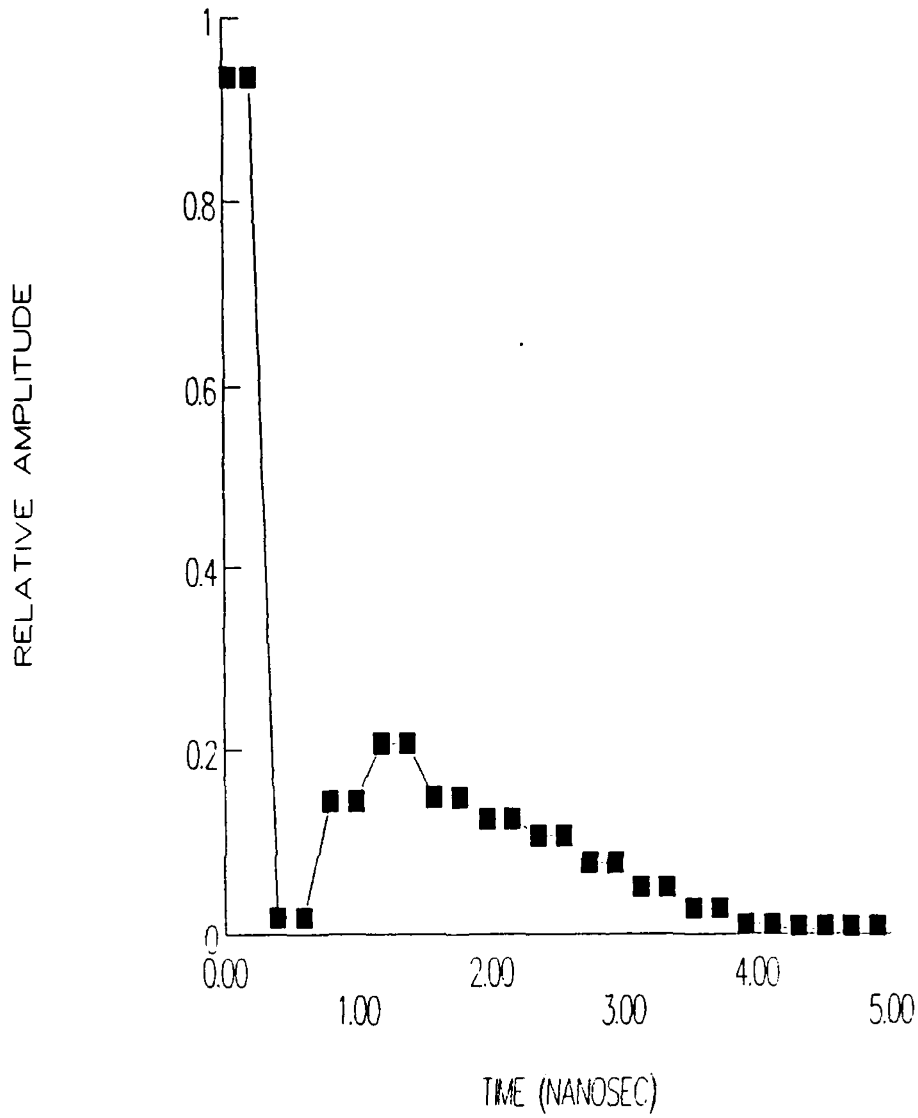


Figure 62. Time Domain Representation of Wide Low-Q Exact E-Pulse

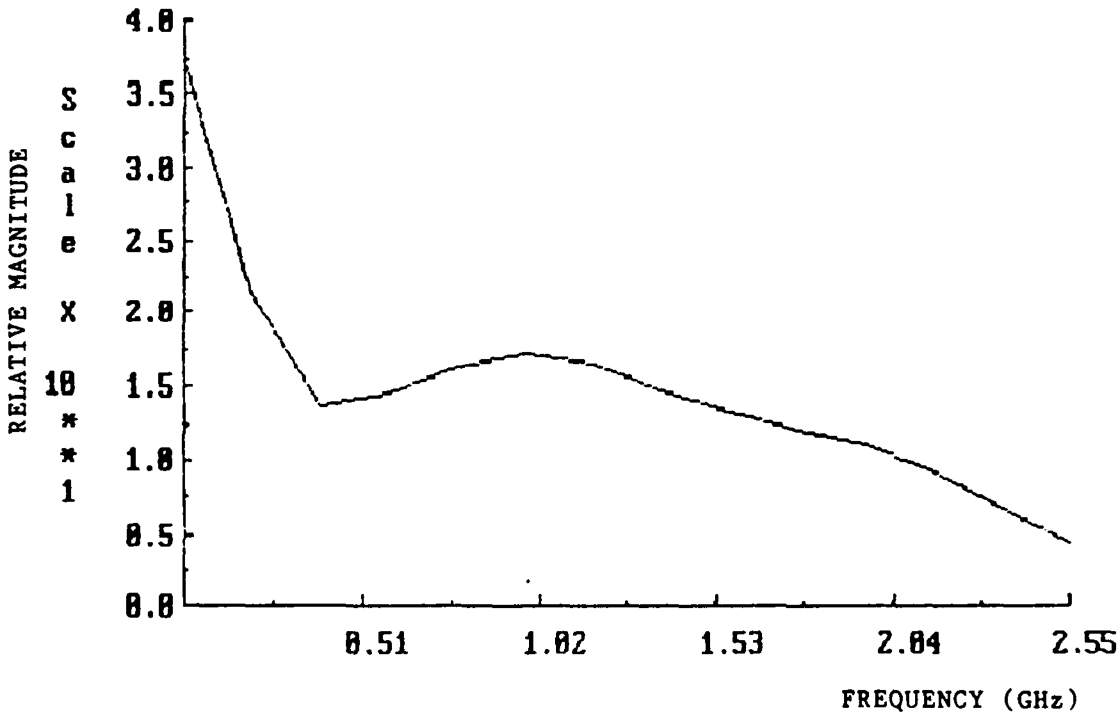


Figure 63. Frequency Response of Wide Low-Q Exact E-Pulse

TABLE 11. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR E-PULSE ($T=\Delta t$) AND HIGH-Q, MEDIUM-Q, AND LOW-Q TEST SIGNALS

Test Signal	E-Pulse ($T=\Delta t$)	SNR			
		∞	30dB	20dB	10dB
	5% above	3.557432ε-08	4.123213ε-07	4.077691ε-06	4.068707ε-05
High-Q	Exact	3.550197ε-08	4.121978ε-07	4.074151ε-06	4.064367ε-05
	5% below	3.586935ε-08	4.125894ε-07	4.071475ε-06	4.060389ε-05
	5% above	1.597413ε-11	4.171517ε-07	4.153723ε-06	4.058114ε-05
Medium-Q	Exact	1.738908ε-15	4.167747ε-07	4.149806ε-06	4.054215ε-05
	5% below	4.046256ε-11	4.165128ε-07	4.146220ε-06	4.050487ε-05
	5% above	2.779841ε-11	4.134989ε-07	4.119730ε-06	4.026625ε-05
Low-Q	Exact	4.026597ε-11	4.134230ε-07	4.119346ε-06	4.026384ε-05
	5% below	2.048559ε-13	4.139902ε-07	4.122647ε-07	4.028809ε-05

When the wider E-Pulse was used, discrimination was greatly improved. The E-Pulse with the exact pole data consistently provided the lowest output energy in the late-time for the high-Q and medium-Q signals, in environments with a SNR as noisy as 10 dB. Data for the high-Q and medium-Q signals is listed in Table 12.

A display of the convolution output for the 10 dB SNR medium-Q signal and the wide exact pole E-Pulse is provided in Figure 64. The discontinuities at the late-time transition are also present in this case, as well as an increase of magnitude at the end of the late-time. These large amplitudes are again due to the loss of the cancellation effect when the entire E-Pulse is not operating on the signal. The frequency spectrum is shown in Figure 65, and illustrates virtually complete extinction of high frequency signal components.

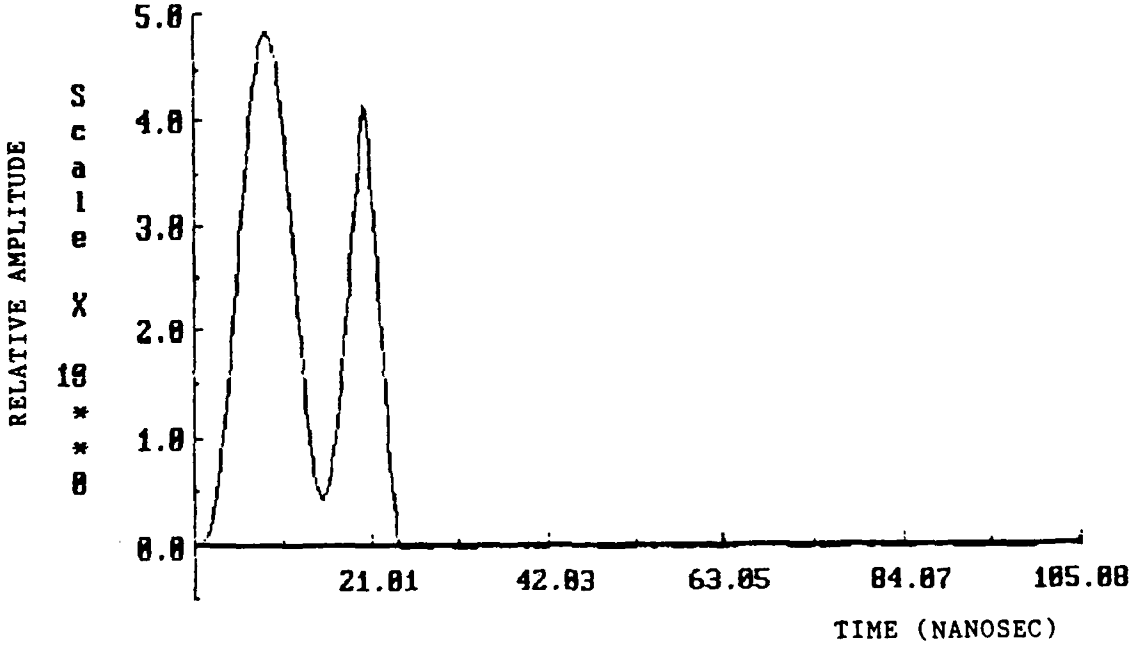


Figure 64. Convolution Output of Medium-Q Test Signal (SNR=10 dB) and Wide Exact E-Pulse

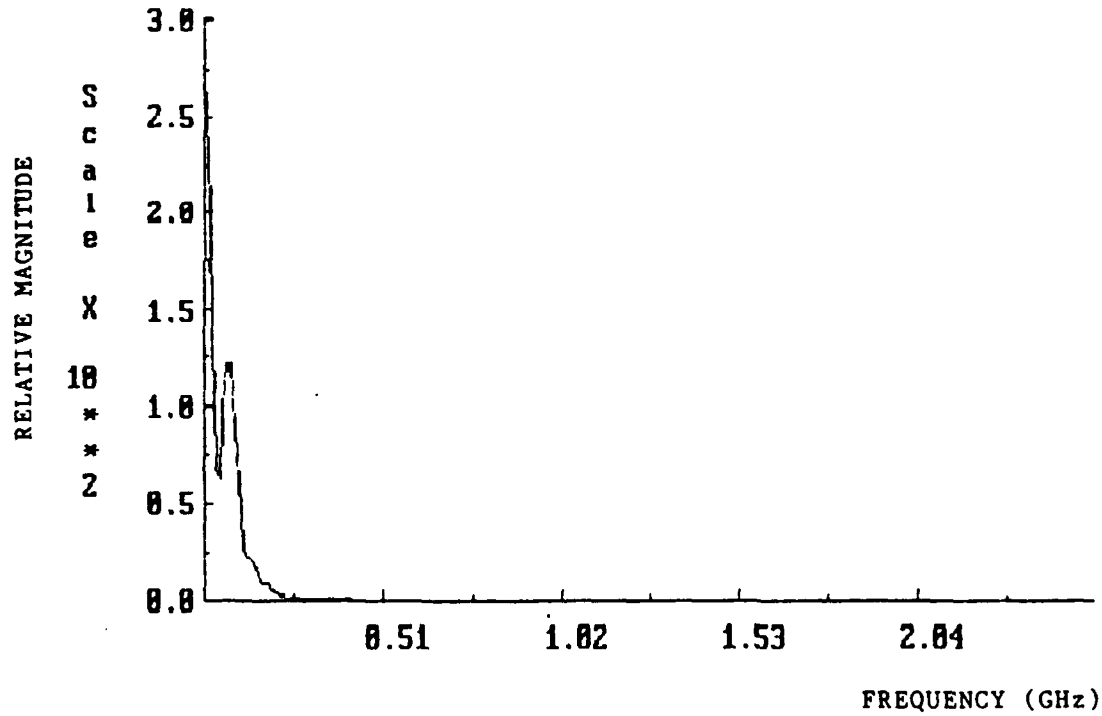


Figure 65. Frequency Spectrum of Convolution Output of Medium-Q Test Signal and Wide Exact E-Pulse

TABLE 12. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR E-PULSE (T=2Δt) AND HIGH-Q AND MEDIUM-Q, TEST SIGNALS

Test Signal	E-Pulse (T = 2Δt)	SNR			
		∞	30dB	20dB	10dB
High-Q	5% above	9.549499ε-05	9.611517ε-05	9.890176ε-05	1.290548ε-04
	Exact	8.726875ε-06	9.353927ε-06	1.322596ε-05	4.752658ε-05
	5% below	3.817654ε-03	1.202017ε-04	1.251210ε-04	1.1614622ε-04
Medium-Q	5% above	2.475081ε-05	2.418240ε-05	2.551004ε-05	5.200848ε-05
	Exact	7.959423ε-15	3.815106ε-07	3.798812ε-06	3.711236ε-05
	5% below	2.8474218ε-05	3.022107ε-05	3.600882ε-05	7.667961ε-05

For the low-Q signal, the correct identification was made down to a SNR of 30 dB. Below that threshold, the E-Pulse did not properly discriminate against the signals that varied five percent from the correct signal. This is considered to be due to the lack of available late-time energy in the low-Q test signal for discrimination and the poor spectral shape of the E-Pulse for the low-Q signal (see Figure 63). Data for the low-Q signal is listed in Table 13.

TABLE 13. CONVOLUTION OUTPUT LATE-TIME ENERGY FOR E-PULSE (T=2Δt) AND LOW-Q TEST SIGNAL

E-Pulse (T = 2Δt)	SNR			
	∞	30dB	20dB	10dB
5% above	5.587498ε-07	1.045437ε-06	6.275899ε-06	5.911904ε-05
Exact	1.542725ε-14	6.783604ε-07	6.754818ε-06	6.600916ε-05
5% below	2.068236ε-07	1.126483ε-06	7.976259ε-06	7.373226ε-05

IV. SUMMARY AND CONCLUSIONS

A. SUMMARY

This thesis has furthered the development of complex resonance annihilation and has compared the effectiveness of the K-Pulse and the E-Pulse as methods of aspect independent target identification. Chapter I introduced the motivation for research in this area, and Chapter II discussed the principles of natural resonance scattering and the resonance annihilation filtering concept. Chapter II also presented the theoretical framework for the development of the K-Pulse and the E-Pulse.

A method for constructing synthetic scattered target return signals, as reported by Jean [Ref. 1], was implemented in FORTRAN for a personal computer. An algorithm to ensure that the proper sampling rate is used in the signal construction has been added, as is discussed in Section IIIA1. Sections IIIA2 and IIIA3 introduced the signals that were to be used in testing the effectiveness of the K-Pulse and the E-Pulse in complex resonance annihilation.

Improving on the work of Dunavin [Ref. 2], the K-Pulse method has been validated using a FORTRAN implementation on a personal computer. This K-Pulse uses a FIR filter with coefficients derived from the z-transform polynomial. Dunavin's effort has been extended to include tests against high, medium, and low-Q synthetic signals under a variety of conditions of noise. The use of a double Gaussian smoothing function in conjunction with the K-

Pulse, as a technique to enhance K-Pulse performance in a noisy environment, was introduced in Section IIIB1.

Results of K-Pulse testing were presented in Sections IIIB2 and IIIB3. The unsmoothed K-Pulse was effective in a noiseless environment, but the requirement for smoothing in noisy conditions is clear, based upon the frequency response of the unsmoothed K-Pulse as shown in Figure 29. When convolved with a double Gaussian smoothing function, the K-Pulse was effective in reducing the effect of noise. When the double Gaussian function was optimized, the K-Pulse was capable of distinguishing high-Q and medium-Q synthetic targets with SNR's as low as 10 dB.

The E-Pulse method of complex resonance annihilation was implemented in FORTRAN for a personal computer using the algorithm that required use of a forcing component. E-Pulses were tested against the identical signals used to test K-Pulses, and the results are presented in Sections IIIC1 and IIIC2.

E-Pulses that were formed using basis functions that were equal in width to the sampling period of the synthetic scattered return signal provided excellent signal cancellation in a noiseless environment. However, the frequency response of this E-Pulse (Figure 47) is identical to the unsmoothed K-Pulse, and the need for a noise reduction scheme for the E-Pulse is evident. The only degree of freedom available to achieve noise reduction in the E-Pulse is the width of the basis function. It was found that using integer multiples (n=2,3,4) of the sampling period of the synthetic scattered return signals to form the width of the basis functions provided a high degree of noise reduction. When the width of the basis functions was optimized, the E-

Pulse was effective in distinguishing high-Q and medium-Q targets in environments with SNR's as low as 10 dB, and a low-Q target under 30 dB SNR conditions.

B. CONCLUSIONS

Both the K-Pulse and the E-Pulse performed well in producing the desired effect of annihilation of the late-time signal response of the synthetic signals in noiseless and noisy environments.

Use of a double Gaussian smoothing function compensated for the monotonically increasing frequency response of the unsmoothed K-Pulse when operating on noisy signals. It was found that the smoothed K-Pulse could be optimized by varying the spreading coefficient of the fast-acting Gaussian component of the double Gaussian function. This resulted in an improved capability to correctly distinguish signals that differed by only five percent in pole composition. The optimum smoothed K-Pulse utilized a double Gaussian smoothing function that, when convolved with the unsmoothed K-Pulse, had a minimum frequency response in the high frequency range and exhibited deep nulls at the frequencies of the scattered signal poles.

E-Pulses constructed using rectangular pulse basis functions of width equal to the sampling period of the scattered signal exhibited the same monotonically increasing frequency response as unsmoothed K-Pulses. This effect was overcome by constructing E-Pulses using rectangular pulse basis functions of widths equal to integer multiples of the sampling period of the scattered signal. It was found that there was an optimum basis function

width (T=2Δt) that would result in an improved capability to distinguish signals that differed by only five percent in pole composition.

Based on the data presented in Tables 7a and 7b for the K-Pulse and in Table 12 for the E-Pulse, both techniques were effective in distinguishing synthetic high-Q and medium-Q targets under conditions with SNR's as low as 10 dB. The E-Pulse also distinguished a synthetic low-Q target in a 30 dB SNR environment (Table 13).

A difference between the ratios of late-time energy of the incorrect filter output to the correct filter output may be used as a measure of effectiveness in comparing the K-Pulse and the E-Pulse. Table 14 is a compilation of these ratios for the high-Q and medium-Q signals. It is observed that the K-Pulse provides better ratios than the E-Pulse for the high-Q signals, but that the E-Pulse provides better ratios than the K-Pulse for the medium-Q signals. Final conclusions on the relative merits of the two methods should not be drawn from this single comparison, and a more detailed evaluation using a variety of signals with different damping coefficients and frequencies should be conducted.

TABLE 14. RATIO OF LATE-TIME ENERGIES: INCORRECT FILTER OUTPUT VS. CORRECT FILTER OUTPUT

RAF	High-Q SNR				Medium-Q SNR			
	∞	30dB	20dB	10dB	∞	30dB	20 dB	10dB
K-Pulse (B=3.0)	22.3:1	18.2:1	18.2:1	7.6:1	704.7:1	41.8:1	5.1:1	1.3:1
E-Pulse T=2Δt	10.9:1	10.3:1	7.5:1	2.4:1	$3.1x10^9$:1	63.6:1	6.7:1	1.4:1

It must be considered that the E-Pulse, as formulated here, has only one degree of freedom to compensate for noise, which is the width of the basis function. The K-Pulse has two degrees of freedom, the width of the double Gaussian smoothing function, and the spacing between the filter weights. This thesis introduced the concept of optimizing the spreading coefficient of the fast-acting Gaussian component of the double Gaussian function. Follow-on efforts may explore the effect of varying the spacing between the filter weights in combination with use of the double Gaussian function. Further comparisons with the E-Pulse are also required using synthetic signals as well as data obtained from test targets in the laboratory.

The failure of the K-Pulse to distinguish the low-Q target and of the E-Pulse to distinguish this target below 30 dB SNR's should not be interpreted as a failure of the techniques. The low-Q target most closely approximates a sphere and is a "worst-case" scenario. This type of signal has very little energy in the late-time and was designed to test the limits of the effectiveness of the K-Pulse and the E-Pulse.

The objectives of this thesis were to convert existing BASIC K-Pulse programs to FORTRAN, develop a FORTRAN implementation of the E-Pulse using a forcing component, test the E-Pulse and K-Pulse against synthetic signals used by Dunavin [Ref. 2], and to test the ability of the K-Pulse and the E-Pulse to distinguish high-Q, medium-Q, and low-Q targets in a variety of conditions of noise. These goals have been achieved, and demonstrate that both the K-Pulse and the E-Pulse have significant potential for implementation in radar target identification schemes and that research in the area of Resonance Annihilation Filtering should continue.

APPENDIX A. SYNTHETIC SIGNAL GENERATOR PROGRAM

A. PROGRAM DESCRIPTION

This program constructs a synthetic scattered return signal. The interactive program format requires the following user specified inputs:

- number of pole pairs in the signal;
- time window (in nanoseconds);
- number of time points;
- late-time start (to define length of early-time sequence);
- damping coefficient, angular frequency, amplitude and phase of each signal pole.

The program first ensures the signal to be entered will be sampled at or above the Nyquist frequency. It then reads the inputs and constructs the early-time and late-time portions of the signal. The early-time component is scaled to match the amplitude of the late-time portion at the late-time start to prevent a sharp discontinuity in the time sequence. The addition of zero-mean, white Gaussian noise, if desired is accomplished by subroutine generation of a randomly distributed Gaussian sequence. The desired SNR is obtained by computing the average power in the late-time portion of the signal and adding the appropriate noise component to each time point of the signal. The entire sequence is then normalized by the RMS energy of the late-time portion, and is stored in a user specified output file.

B. PROGRAM LISTING

A listing of the synthetic signal generator program is included below.

```
C       PROGRAM SIG.FOR
```

```
C
C     PROGRAM NOTES:
C          1) PROGRAM GENERATES SYNTHETIC SIGNAL
C          2) KEYBOARD INPUTS: POLES/REAL,IMAG,MAGNITUDE,PHASE/
C          3) OUTPUT TO DATA FILE:SYNTHETIC SIGNAL TIME SERIES
C          4) GAUSSIAN NOISE ADDITION TO SYN SIGNAL OPTION
C          5) PROGRAM REQUIRES 8087 MATH CO-PROCESSOR TO RUN
C          6) ORIGINAL SYNGEN PROGRAM WRITTEN IN BASIC
C             BY DR. M. A. MORGAN
C          7) STWO87FOR PROGRAM WRITTEN IN MSFORTRAN 77
C             BY L. A. CHEEKS
C          8) UPDATE AUG89 BY LT M.S. SIMON
C
C     ***DECLARATIONS***
C
      REAL T9,T0,A,C9,PI,X0SUM,GSUM,GSUMPRM,R,CDB,X0(1024),R0(100),I0(10
     +0),A0(100),P0(100),G(1024),X1(1024),P,TZ/0.0/,DT1,DT,L2,X02SUM,X02
     +(1024)
      INTEGER N,Z,I9,I1,NCOUNT
      CHARACTER FN*8,TITLE*64,ANS*1
      DATA PI/3.1415927/
C
C     ***MAIN BODY OF PROGRAM***
C
C     ***DATA INPUT ROUTINE***
C
 1    WRITE(*,*) 'SYNTHETIC SIGNAL GENERATOR'
      CALL NYQUIST(DT1)
 2    WRITE(*,*)' '
      WRITE(*,*)'CHOOSE LENGTH OF TOTAL TIME SERIES AND TOTAL NUMBER'
      WRITE(*,*)'OF SAMPLE POINTS TO ENSURE DELTA T < ',DT1,' NANOSEC'
      WRITE(*,*)' '
      WRITE(*,*) 'ENTER THE LENGTH OF THE TOTAL TIME SERIES(NANOSEC)'
      READ(*,*) T9
      WRITE(*,*) 'LENGTH OF THE TOTAL TIME SERIES = ',T9,' NANOSEC'
      WRITE(*,*) 'ENTER THE TOTAL NUMBER OF SAMPLE POINTS  (**2)'
      READ(*,240) N
      WRITE(*,*) 'TOTAL NUMBER OF SAMPLE POINTS = ',N
      WRITE(*,*) 'LENGTH OF THE TOTAL TIME SERIES = ',T9,' NANOSEC'
      DT=T9/N
      IF(DT.GT.DT1)THEN
         WRITE(*,*)'DELTA T IS TOO LARGE. REVISE LENGTH OF TIME SERIES'
         WRITE(*,*)'AND TOTAL NUMBER OF SAMPLE POINTS'
         GO TO 2
      ELSE
         GO TO 3
      ENDIF
3     CONTINUE
      WRITE(*,*) 'ENTER THE LATE TIME START (NANOSEC)'
      READ(*,*) T0
      WRITE(*,*) 'LATE TIME START = ',T0,' NANOSEC'
      WRITE(*,*) 'ENTER NUMBER OF POLE PAIRS IN LATE-TIME'
      READ(*,*) Z
      WRITE(*,*) 'NUMBER OF POLE PAIRS IN LATE-TIME =  ',Z
12    DO 20 I=1,Z
```

```
        WRITE(*,*) 'ENTER REAL PART OF POLE #',I, '  IN NEP/NANOSEC'
        READ(*,*) R0(I)
        WRITE(*,*) 'ENTER IMAG PART OF POLE #',I, '   IN RADIANS/NANO
     +SEC'
        READ(*,*) I0(I)
        WRITE(*,*) 'ENTER MAGNITUDE OF POLE #',I, '  IN VOLTS'
        READ(*,*) A0(I)
        WRITE(*,*) 'ENTER PHASE OF POLE    #',I, '  IN RADIANS'
        READ(*,*) P0(I)
20    CONTINUE
22    WRITE(*,*) 'ENTER OUTPUT FILE NAME (MAX 8 CHAR)'
      READ(*,270) FN
      WRITE(*,*) 'OUTPUT FILE NAME IS ',FN
      WRITE(*,*) 'ENTER OUTPUT TITLE (MAX 64 CHAR)'
      READ(*,280) TITLE
      WRITE(*,*) 'OUTPUT TITLE IS ',TITLE
C
C     ***INPUT DATA CHECK***
C
      WRITE(*,*) 'INPUT DATA CHECK'
      WRITE(*,*) 'OUPUT FILE NAME IS            ',FN
      WRITE(*,*) 'OUTPUT TITLE IS               ',TITLE
      WRITE(*,*) 'NUMBER OF POLE PAIRS        = ',Z
      WRITE(*,*) 'NUMBER OF SAMPLE POINTS     = ',N
      WRITE(*,*) 'LENGTH OF TOTAL TIME SERIES = ',T9, ' NANOSEC'
      WRITE(*,*) 'EARLY TIME INTERVAL         = ',T0, ' NANOSEC'
      DO 50 I=1,Z
        WRITE(*,*) 'REAL PART OF POLE #',I,' = ',R0(I),' NEP/NANOSEC'
        WRITE(*,*) 'IMAG PART OF POLE #',I,' = ',I0(I),'  RADIANS/NAN
     +OSEC'
        WRITE(*,*) 'MAGNITUDE OF POLE #',I,' = ',A0(I),' VOLTS'
        WRITE(*,*) 'PHASE OF POLE #    ',I,' = ',P0(I),' RADIANS'
        PAUSE
50    CONTINUE
55    WRITE(*,*) 'IS DATA CORRECT?'
      READ(*,260) ANS
      IF((ANS.EQ.'Y').OR.(ANS.EQ.'y'))THEN
        WRITE(*,*) 'DATA IS CORRECT'
        GO TO 70
      ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n'))THEN
        WRITE(*,*) 'DATA ERRORS. PLEASE START AGAIN.'
        GO TO 01
      ELSE
        WRITE(*,*) 'TRY AGAIN! Y OR N'
        GO TO 55
      ENDIF
C
C     ***CALCULATIONS***
C
70    WRITE(*,*) 'CALCULATIONS IN PROGRESS - A'
      A=3.0*PI/2.0
      I9=0
      X0SUM=0.0
      NCOUNT=0
      DT=T9/(N-1)
```

```
      DO 80 I=1,N
           T=(I-1)*DT
           IF(T.GE.T0) GO TO 72
           I9=I
           GO TO 80
72         X0(I)=0.0
           DO 78 I1=1,Z
               X0(I)=X0(I)+A0(I1)*EXP((T-T0)*R0(I1))*COS((T-T0)*I0(I1)+
     +P0(I1))
           NCOUNT=NCOUNT+1
78         CONTINUE
      X0SUM=X0SUM+(X0(I)**2)
80    CONTINUE
C     X0SUM/NCOUNT IS AVERAGE POWER IN LATE TIME INTERVAL
      X0SUM=X0SUM/NCOUNT
      WRITE(*,*) 'CALCULATIONS IN PROGRESS - B'
      IF(I9.EQ.0) GO TO 100
      C9=X0(I9+1)
      DO 90 I=1,I9
           T=(I-1)*DT
           X0(I)=C9*SIN(A*T/T0)**2
90    CONTINUE

100   WRITE(*,*) 'CALCULATIONS COMPLETE'
C
C     ***ROUTINE TO ADD GAUSSIAN NOISE TO SIGNAL***
C
      GSUM=0.0
      WRITE(*,*) 'DO YOU WISH TO ADD GAUSSIAN NOISE TO THE SIGNAL?'
108   READ(*,260) ANS
      IF((ANS.EQ.'Y').OR.(ANS.EQ.'y'))THEN
           WRITE(*,*) 'OKAY'
           GO TO 120
      ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n'))THEN
           WRITE(*,*) 'OKAY.  NO NOISE'
           GO TO 155
      ELSE
           WRITE(*,*) 'TRY AGAIN! Y OR N'
           GO TO 108
      ENDIF
120   WRITE(*,*)'ENTER SEED FOR RANDOM NUMBER GENERATION FOR NOISE'
      WRITE(*,*)'(0 < R < 1.0)  0.21289 GENERATES A GOOD DISTRIBUTION'
      READ(*,250) R
      DO 130 I=1,N
           CALL NORNG(R,P)
           G(I)=P
           GSUM=GSUM+(G(I)**2)
130   CONTINUE
C     GSUM/N IS AVERAGE POWER OF GAUSSIAN SIGNAL
      GSUM=GSUM/N
      WRITE(*,*) 'ENTER DESIRED SIGNAL TO NOISE RATIO IN DB'
      READ(*,*) DB
      GSUMPRM=X0SUM/(10.0**(DB/10.0))
      CDB=10.0*ALOG10(X0SUM/GSUMPRM)
      WRITE(*,*) 'CHECK DB =',CDB
```

```
      RATIO=SQRT(GSUMPRM/GSUM)
      DO 140 I=1,N
           G(I)=RATIO*G(I)
140   CONTINUE
      DO 150 I=1,N
           X0(I)=X0(I)+G(I)
150   CONTINUE
C
C     ***ROUTINE TO NORMALIZE SYN GEN SIGNAL***
C
155   X02SUM=0.0
      DO 156 I=I9+1,N
         X02(I)=X0(I)**2
         X02SUM=X02SUM+X02(I)
156   CONTINUE
      L2=SQRT(X02SUM)
      DO 160 I=1,N
         X0(I)=X0(I)/L2
160   CONTINUE
C
C
C     ***ROUTINE TO STORE SYN GEN SIGNAL DATA IN DATA FILE***
C
      WRITE(*,*) 'STORING SYN GEN SIGNAL DATA IN DATA FILE'
      OPEN(1,FILE=FN)
      WRITE(1,280) TITLE
      WRITE(1,240) N
      WRITE(1,250) TZ
      WRITE(1,250) T9
      DO 190 I=1,N
           WRITE(1,250) X0(I)
190   CONTINUE
      CLOSE(1)
      WRITE(*,*) 'SYN GEN SIGNAL STORED IN ',FN
C
C     ***ROUTINE TO DO ANOTHER RUN***
C
      WRITE(*,*) 'GENERATE SAME SIGNAL WITH A DIFFERENT SNR?'
195   READ(*,260) ANS
      IF((ANS.EQ.'Y').OR.(ANS.EQ.'y'))THEN
           WRITE(*,*) 'OK.  SAME SIGNAL WITH A DIFFERENT SNR'
           GO TO 22
      ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n'))THEN
           WRITE(*,*) 'OK'
           GO TO 210
      ELSE
           WRITE(*,*) 'TRY AGAIN! Y OR N'
           GO TO 195
      ENDIF
210   WRITE(*,*) 'HOW ABOUT A DIFFERENT SIGNAL (NEW POLES)?'
212   READ(*,260) ANS
      IF((ANS.EQ.'Y').OR.(ANS.EQ.'y'))THEN
           WRITE(*,*) 'OK.'
           WRITE(*,*) 'CHOOSE ONE OF THE FOLLOWING. A OR B'
           WRITE(*,*) 'A - CHANGE JUST POLE DATA; LEAVE REST ALONE'
```

```
      WRITE(*,*) 'B - CHANGE ALL DATA'
222   READ(*,260) ANS
      IF((ANS.EQ.'A').OR.(ANS.EQ.'a'))THEN
           WRITE(*,*) 'OK - CHANGE JUST POLE DATA'
           GO TO 12
      ELSEIF((ANS.EQ.'B').OR.(ANS.EQ.'b'))THEN
           WRITE(*,*) 'OK - CHANGE ALL THE DATA'
           GO TO 01
      ELSE
           WRITE(*,*) 'TRY AGAIN! A OR B'
           GO TO 222
      ENDIF
 ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n'))THEN
      WRITE(*,*) 'OK'
      GO TO 230
 ELSE
      WRITE(*,*) 'TRY AGAIN! Y OR N'
      GO TO 212
 ENDIF
230   WRITE(*,*) 'SYN GEN PROGRAM IS COMPLETE'
240   FORMAT(I5)
250   FORMAT(E12.6)
260   FORMAT(A1)
270   FORMAT(A8)
280   FORMAT(A64)
      WRITE(*,*) 'CHECK NEW FILE(S)'
      STOP
      END
C
C     ***SUBROUTINES***
C
      SUBROUTINE NORNG(R,P)
C
C     THIS SUBROUTINE GENERATES A SEQUENCE OF NUMBERS NORMALLY
C     AND RANDOMLY DISTRIBUTED OVER THE INTERVAL -3 TO 3 FROM
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS BY THE METHOD OF LINEAR
C     APPROXIMATION TO THE INVERSE OF THE ACCUMULATIVE
C     NORMAL DISTRIBUTION FUNCTION.
C     REF: P.282 OF FORTRAN: Scientific Subroutine Library
C
C     NOTES:
C          1) R = 0.21289 GENERATES A GOOD NORMAL DISTRIBUTION
C
      DIMENSION Y(6),X(6),S(5)
      DATA Y/0.0,0.0228,0.0668,0.1357,0.2743,0.5/
      DATA X/-3.01,-2.0,-1.5,-1.0,-0.6,0.0/
      DATA S/43.8596,11.3636,7.25689,2.891352,2.65887/
      CALL STRNUM(R)
      P=R
      I=1
      IF(P.GT.(0.5)) P=1.0-R
2     IF(P.LT.Y(I+1)) GO TO 8
      I=I+1
      GO TO 2
8     P=((P-Y(I))*S(I)+X(I))
```

```
      IF(R.GE.(0.5)) P=-P
      RETURN
      END
      SUBROUTINE STRNUM(R)
C
C     THIS SUBROUTINE GENERATES A SEQUENCE OF NUMBERS WHICH ARE
C     RANDOMLY AND UNIFORMLY DISTRIBUTED OVER THE UNIT INTERVAL.
C     REF: P. 280 OF FORTRAN: Scientific Subroutine Library
C
C     NOTES:
C          1) R = 0.21289 GENERATES A GOOD UNIFORM DISTRIBUTION
C
      BB=1.0
      P1=R*317.0
      R=AMOD(P1,BB)
      RETURN
      END
C
C     *****SUBROUTINE NYQUIST*****
C        LT M.S. SIMON 8/10/89
C
C
      SUBROUTINE NYQUIST(DT1)
C
C     *****DECLARATIONS*****
C
      REAL E,A,R,DT1,LP,WN,WF,NF
C
C     *****INPUTS*****
C
      WRITE(*,*)' '
      WRITE(*,*)'ROUTINE TO CALCULATE NYQUIST FREQUENCY'
      WRITE(*,*)'ENTER ABSOLUTE VALUE FOR LARGEST POLE'
      READ(*,*)LP
      WRITE(*,*)'ENTER VALUE OF LARGEST OMEGA'
      READ(*,*)WN
      WRITE(*,*)'ENTER VALUE IN DB FOR CUTOFF'
      READ(*,*)C
C
C     *******CALCULATIONS*****
C
      PI=3.1415927
      E=10**(C/10)
      A=SQRT(1/E-1)
      R=LP*A
      WF=WN+R
      NF=WF/PI
      DT1=1/NF
      RETURN
      END
```

APPENDIX B. K-PULSE GENERATOR

A. PROGRAM DESCRIPTION

This program constructs the unsmoothed K-Pulses that are used to annihilate input target return signals. The interactive format of the program requires the following inputs:

- the number of pole pairs for the K-Pulse
- the scale factor (for scaled targets);
- the K-Pulse time window (in nanoseconds);
- the number of time points;
- the number of time steps desired between K-Pulse filter weights.;
- the damping coefficient and the angular frequency of each signal pole.

The program reads the inputs and scales the poles. A subroutine computes the filter coefficients by forming the z-transform polynomial, and the coefficients are then normalized by the RMS energy of the K-Pulse and stored in a user specified output file.

B. PROGRAM LISTING

A listing of the K-Pulse generator program is included below.

```
      PROGRAM KP
C
C        PROGRAM NOTES
C             1) PROGRAM GENERATES KILL PULSES
C             2) KEYBOARD INPUTS:  POLES/REAL,IMAG
C             3) OUTPUT TO DATA FILE:  K-PULSE TIME SERIES
C             4) PROGRAM REQUIRES 8087 MATH CO-PROCESSOR
C             5) PROGRAM WRITTEN IN MSFORTRAN BY PROF. M.A. MORGAN
C             6) INTERACTIVE MODS BY LT M.S. SIMON
C
C     K-Pulse Generation Using Z-Transform Product Non-Recursive
C     Filter Design Approach by M.A. Morgan 8/24/89.
C
```

```
      REAL SR(20),SI(20),A(0:100),B(2),H(401),SF
      CHARACTER FNAME*16,TITLE*64,ANS*1
      WRITE(*,*) '*** K-PULSE GENERATOR ***'
      WRITE(*,*)
 1    WRITE(*,*) 'ENTER NUMBER OF POLE PAIRS (<=50):'
      READ(*,*) NP
        WRITE(*,*) 'ENTER SCALE FACTOR'
        READ(*,*) SF
      WRITE(*,*) 'ENTER TIME WINDOW (NANOSEC):'
      READ(*,*) T0
      WRITE(*,*) 'ENTER NUMBER OF TIME POINTS:'
      READ(*,*) NT
            DT=T0/(NT-1.0)
      WRITE(*,*) 'ENTER TIME-STEPS BETWEEN FILTER WEIGHTS (>=1):'
      READ(*,*) KS
        DO 11 N=1,NP
      WRITE(*,*) 'ENTER SIGMA (GigaNep/Sec < 0 ) FOR POLE #',N
      READ(*,*) SR(N)
      WRITE(*,*) 'ENTER OMEGA (GigaRad/Sec > 0 ) FOR POLE #',N
        READ(*,*) SI(N)
 11     CONTINUE
C
C       INPUT DATA CHECK
C
        WRITE(*,*) ' '
        WRITE(*,*)'INPUT DATA CHECK'
        WRITE(*,*)'NUMBER OF POLE PAIRS IS ',NP
        WRITE(*,*)'SCALE FACTOR IS ',SF
        WRITE(*,*)'TIME WINDOW IS ',T0,' NANOSEC'
        WRITE(*,*)'NUMBER OF TIME POINTS IS ',NT
        WRITE(*,*)'NUMBER OF TIME STEPS BETWEEN FILTER WEIGHTS IS ',KS
        WRITE(*,*)' '
        DO 12 N=1,NP
        WRITE(*,*)'SIGMA ',N,' = ',SR(N)
        WRITE(*,*)'OMEGA ',N,' = ',SI(N)
        PAUSE
 12     CONTINUE
 13     WRITE(*,*)'IS DATA CORRECT?'
        READ(*,200)ANS
        IF((ANS.EQ.'Y').OR.(ANS.EQ.'y'))THEN
        WRITE(*,*)'DATA CORRECT'
        GO TO 14
        ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n'))THEN
        WRITE(*,*)'DATA ERRORS. START AGAIN'
        GO TO 1
        ELSE
        WRITE(*,*)'TRY AGAIN! Y OR N'
        GO TO 13
        ENDIF
C
C       SCALING POLES
C
 14     DO 15 N=1,NP
        SR(N)=SR(N)/SF
        SI(N)=SI(N)/SF
```

```
  15     CONTINUE
C
C        CALCULATIONS
C
      MP=0
      DO 22 N=1,NP
      E=EXP(KS*SR(N)*DT)
      C=COS(KS*SI(N)*DT)
      B(1)=-2.0*E*C
      B(2)=E*E
      CALL POLY(A,B,MP)
  22  CONTINUE
      WRITE(*,*)
      WRITE(*,*) 'UN-NORMALIZED K-PULSE FILTER WEIGHTS:'
      WRITE(*,*)
      DO 33 M=0,MP
      WRITE(*,*) M,A(M)
  33  CONTINUE
            PAUSE
C
C     NORMALIZE K-PULSE TO UNIT ENERGY
C
      ENERGY=A(0)**2
      M1=KS*MP+1
      DO 44 M=1,MP
  44  ENERGY=ENERGY+A(M)**2
      RMS=SQRT(ENERGY)
C
C     DEFINING UNIT SAMPLE RESPONSE
C
      H(1)=A(0)/RMS
      I=1
            DO 55 N=1,MP
      M=0
  50    M=M+1
      I=I+1
      IF(M.GE.KS) GO TO 55
      H(I)=0.0
      GO TO 50
  55  H(I)=A(N)/RMS
C
C       STORING K-PULSE IN DATA FILE
C
      TS=0.0
      TF=KS*MP*DT
      WRITE(*,*) 'STORING K-PULSE IN DATA FILE'
      WRITE(*,*)
      WRITE(*,*) 'ENTER OUTPUT FILE (<= 16 CHAR):'
      READ(*,100) FNAME
        WRITE(*,*) 'ENTER HEADER FOR OUTPUT FILE (<= 64 Char):'
      READ(*,100) TITLE
            OPEN(1,FILE=FNAME)
            WRITE(1,100) TITLE
            WRITE(1,110) M1
            WRITE(1,120) TS
```

```
          WRITE(1,120) TF
          DO 66 N=1,M1
      WRITE(1,120) H(N)
 66     CONTINUE
 88     WRITE(*,*)'K-PULSE GENERATOR PROGRAM COMPLETE'
        WRITE(*,*)'CHECK NEW FILE'
100   FORMAT(A)
110       FORMAT(I5)
120   FORMAT(E12.6)
200     FORMAT(A1)
      STOP
      END

      SUBROUTINE POLY(A,B,N)
C
C     MULTIPLYING  {z**2 + B(1)*z + B(2)} x
C     {z**N + A(1)*z**(N-1) + A(2)*z**(N-2) + .... + A(N)} =
C     z**(N+2) + C(1)*z**(N+1) + C(2)*z**N + ....  + C(N+2)
C
C     COMPUTING C(N) COEFFICIENTS AND STORING INTO A(N) WHILE
C     INCREMENTING N --> N + 2
C
      REAL A(0:100),B(2),C(0:100)
      A(0)=1.0
C     INITIALIZE ON FIRST CALL TO ROUTINE
      IF(N.GE.2) GO TO 11
      N=2
      A(1)=B(1)
      A(2)=B(2)
      GO TO 44
 11   C(1)=A(1)+B(1)
      DO 22 I=2,N
 22   C(I)=A(I)+A(I-1)*B(1)+A(I-2)*B(2)
      C(N+1)=A(N)*B(1)+A(N-1)*B(2)
      C(N+2)=A(N)*B(2)
      N=N+2
      DO 33 I=1,N
 33   A(I)=C(I)
 44   CONTINUE
      RETURN
      END
```

APPENDIX C. DOUBLE GAUSSIAN SMOOTHING FUNCTION GENERATOR

A. PROGRAM DESCRIPTION

This program generates the double Gaussian smoothing function used to enhance K-Pulse performance in noisy environments. The interactive format of the program requires the following inputs:

- number of time points, time window, number of pole pairs, number of time steps, and number of time steps between filter weights used in forming the K-Pulse to be smoothed (this permits the smoothing function to be matched to the K-Pulse);
- the smallest value of angular frequency, ω, used in forming the K-Pulse;
- the spreading coefficient for the fast-acting Gaussian component used to generate the desired width of the smoothing function;
- the value of the slow-acting Gaussian component required to truncate the smoothing function.

The program computes the smoothing function and then truncates the sequence in accordance with the specified value. A Hamming window is then applied to the truncated sequence to reduce the magnitude of the high frequency components in its frequency response. The smoothing function values are then stored in a user specified output file.

B. PROGRAM LISTING

A listing of the double Gaussian smoothing function generator program is included below.

```
C       PROGRAM DBLGSS.FOR
C
C       PROGRAM NOTES:
C           1) PROGRAM GENERATES DOUBLE GAUSSIAN SMOOTHING FOR K-PULSES
```

```
C          2) KEYBOARD INPUTS: K-PULSE DATA
C          3) OUPUT TO DATA FILE: DOUBLE GAUSSIAN SMOOTHING FUNCTION
C          4) PROGRAM REQUIRES 8087 MATH CO-PROCESSOR TO RUN
C          5) ORIGINAL WRITTEN IN BASIC BY DR. M. A. MORGAN
C          7) MODIFIED FOR MSFORTRAN AUG89 BY LT M.S. SIMON
C
C     DECLARATIONS
      REAL T9,ALPHA1,ALPHA2,TAU,A1,A2,LQ,DT,F(1024),AAA1,AAA2,TZ/0.0/,
     +WMIN,B,TF,T,G(1024),R,THRESH,W(1024)
      INTEGER Z,N,NSUM,LP,K,M,RZ,I,L,C
      CHARACTER FN*8,TITLE*64
C
C     ***MAIN BODY OF PROGRAM***
C
C     ***DATA INPUT ROUTINE***
C
      WRITE(*,*)'DOUBLE GAUSSIAN SMOOTHING FUNCTION GENERATOR'
      WRITE(*,*)' '
      WRITE(*,*)'ENTER NUMBER OF TIME POINTS USED TO GENERATE K-PULSE'
      READ(*,*)N
      WRITE(*,*)'ENTER TIME WINDOW (NANOSEC) USED FOR K-PULSE'
      READ(*,*)T9
      WRITE(*,*)'ENTER NUMBER OF POLE PAIRS USED IN FORMING K-PULSE'
      READ(*,*)Z
      WRITE(*,*)'ENTER NUMBER OF TIME STEPS BETWEEN K-PULSE FILTER WTS'
      READ(*,*)KS
      WRITE(*,*)'ENTER SMALLEST VALUE OF OMEGA USED TO FORM K-PULSE'
      READ(*,*)WMIN
      WRITE(*,*)'ENTER COEFFICIENT OF DOUBLE GAUSSIAN'
      READ(*,*)B
      WRITE(*,*)'ENTER VALUE FOR THRESHOLD'
      READ(*,*)C
      WRITE(*,*)'ENTER OUTPUT FILE NAME (MAX 8 CHAR)'
      READ(*,30)FN
      WRITE(*,*)'ENTER OUTPUT TITLE'
      READ(*,40)TITLE
C
C     ***ROUTINE TO FORM DOUBLE-GAUSSIAN SMOOTHING FUNCTION***
C
      DT=T9/(N-1)
      NSUM=Z*KS
      LP=2*NSUM+1
      WRITE(*,*) 'FORMING SMOOTHING FUNCTION'
      ALPHA1=B*WMIN/(2.0*SQRT(ALOG(10.0)))
      ALPHA2=WMIN/(2.0*SQRT(ALOG(10.0)))
      TAU=SQRT(ALOG(1000000.0))/ALPHA2
      A2=ALPHA2/(ALPHA1-ALPHA2)
      A1=A2+1.0
      LQ=LP+TAU/DT
      THRESH=C*ALOG(10.0)
      I=0
           DO 10 K=1,N
                T=(K-1.0-LQ)*DT
                AAA1=(ALPHA1*T)**2
                AAA2=(ALPHA2*T)**2
```

```
                IF(AAA1.GT.85.0)THEN
                     AA1=0.0
                ELSE
                     AA1=A1*EXP(-AAA1)
                ENDIF
                IF(AAA2.GT.85.0)THEN
                     AA2=0.0
                ELSE
                     AA2=A2*EXP(-AAA2)
                ENDIF
                F(K)=AA1-AA2
                IF(AAA2.GT.THRESH)THEN
                GO TO 9
                ELSE
                I=I+1
                G(I)=F(K)
                ENDIF
 9              CONTINUE
10         CONTINUE
        PI=3.1415927
C
C       ***MULTIPLYING SMOOTHING FUNCTION BY HAMMING WINDOW***
C
        WRITE(*,*)'APPLYING HAMMING WINDOW'
         DO 12 K=1,I
           W(K)=0.54-(0.46*COS((2*PI*K)/I))
           G(K)=W(K)*G(K)
12       CONTINUE
        WRITE(*,*)'CALCULATIONS COMPLETE'
        TF=DT*(I-1)
C
C       ***ROUTINE TO STORE SMOOTHING FUNCTION IN DATA FILE***
C
        WRITE(*,*)'STORING SMOOTHING FUNCTION IN DATA FILE'
        OPEN(1,FILE=FN)
        WRITE(1,40) TITLE
        WRITE(1,50) I
        WRITE(1,60) TZ
        WRITE(1,60) TF
        DO 20 K=1,I
           WRITE(1,60) G(K)
20      CONTINUE
        CLOSE(1)
        WRITE(*,*)'SMOOTHING FUNCTION STORED IN FILE ',FN
30      FORMAT(A8)
40      FORMAT(A64)
50      FORMAT(I5)
60      FORMAT(E12.6)
        STOP
        END
```

APPENDIX D. CONVOLUTION PROGRAM TO FORM SMOOTHED K-PULSES

A. PROGRAM DESCRIPTION

This program performs the time convolution of an unsmoothed K-Pulse and a double Gaussian smoothing function, and normalizes the resultant smoothed K-Pulse by its RMS energy.

The program reads the input signals from two files and stores them in arrays. A time convolution without zero padding is performed. Normalization is required to permit the K-Pulse to later be convolved with scattered return signals for late-time energy calculations. The smoothed K-Pulse is stored in a user specified output file.

B. PROGRAM LISTING

The convolution program to form smoothed K-Pulses is included below.

```
      PROGRAM CONK
C
C     NOTES:
C
C           1) PROGRAM CONVOLVES GAUSSIAN SMOOTHING FUNCTION AND
C              UNSMOOTHED K-PULSE
C           2) INPUTS: TIME SERIES FROM TWO DATA FILES
C           3) SMOOTHED K-PULSE OUPUT TO DATA FILE
C           5) ORIGINAL WRITTEN 8/87 IN FORTRAN 77 BY LT L. CHEEKS
C           6) UPDATED MAY 89 BY LT M.S. SIMON & PROF M.A. MORGAN
C
C     DECLARATIONS
C
      REAL X(1024),Y(1024),Z(2048),ENERGY,RMS
      CHARACTER*1 ANS, BELL
      CHARACTER*8 FNX,FNY,FNZ
      CHARACTER*64 TITX,TITY,TITZ
C
      BELL=CHAR(7)
C
C     DATA FILE INPUT
```

```
C
  10  CONTINUE
      WRITE(*,*) 'THIS PROGRAM CONVOLVES X(n) * Y(n) = Z(n)'
      WRITE(*,*) 'AND PLACES Z(n) IN USER SPECIFIED OUTPUT FILE'
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF X(n) INPUT FILE(DOUBLE GAUSSIAN)'
      WRITE(*,*)
      READ(*,100) FNX
      WRITE(*,*)
      WRITE(*,*) 'X(n) INPUT FILE IS ',FNX
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF Y(n) INPUT FILE (UNSMOOTHED K-PULSE)'
      WRITE(*,*)
      READ(*,100) FNY
      WRITE(*,*)
      WRITE(*,*) 'Y(n) INPUT FILE IS ',FNY
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF Z(n) FILE (SMOOTHED K-PULSE)'
      WRITE(*,*)
      READ(*,100) FNZ
      WRITE(*,*)
      WRITE(*,*) 'OUTPUT FILE IS ',FNZ
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER HEADER TITLE FOR Z(n) FILE'
      WRITE(*,*)
      READ(*,100) TITZ
      WRITE(*,*)
C
C     READING INPUT FILES
C
      OPEN(1,FILE=FNX)
      OPEN(2,FILE=FNY)
      OPEN(3,FILE=FNZ)
      READ(1,100) TITX
      READ(1,120) NX
      READ(1,130) TX1
      READ(1,130) TX2
      DO 20 I=1,NX
           READ(1,130) X(I)
  20  CONTINUE
      WRITE(*,*) 'X(t) FILE READ'
      READ(2,100) TITY
      READ(2,120) NY
      READ(2,130) TY1
      READ(2,130) TY2
      DO 30 I=1,NY
           READ(2,130) Y(I)
  30  CONTINUE
      WRITE(*,*) 'Y(t) FILE READ'
      WRITE(*,*)
      CLOSE(1)
```

```
      CLOSE(2)
C
C     CONVOLUTION ALGORITHM WITHOUT ZERO PADDING
C     Written by M.A. Morgan 5/19/89
C
      WRITE(*,*) 'CONVOLUTION IN PROGRESS'
      WRITE(*,*)
C
      DO 50 N=1,NY
      Z(N)=0.0
      M2=MIN0(N,NX)
      DO 40 M=1,M2
   40 Z(N)=Z(N)+X(M)*Y(N-M+1)
   50 CONTINUE
      N1=NY+1
      N2=NX+NY-1
      DO 70 N=N1,N2
      Z(N)=0.0
      M1=N+1-NY
      M2=MIN0(N,NX)
      DO 60 M=M1,M2
   60 Z(N)=Z(N)+X(M)*Y(N-M+1)
   70 CONTINUE
C
C     ROUTINE TO NORMALIZE SMOOTHED K-PULSE
C
   83 WRITE(*,*)'NORMALIZING SMOOTHED K-PULSE'
      ENERGY = Z(1)**2
      DO 85 N=2,N2
      ENERGY = ENERGY + Z(N)**2
   85 CONTINUE
      RMS = SQRT(ENERGY)
      DO 87 N=1,N2
      Z(N)=Z(N)/RMS
   87 CONTINUE
C
C     ROUTINE TO SAVE SMOOTHED K-PULSE TO FILE
C
      WRITE(*,*) 'SAVING SMOOTHED K-PULSE TO OUTPUT DATA FILE'
      WRITE(3,100) TITZ
      WRITE(3,120) N2
      TZ1=TX1+TY1
      TZ2=TX2+TY2
      WRITE(3,130) TZ1
      WRITE(3,130) TZ2
      DO 80 I=1,N2
           WRITE(3,130) Z(I)
   80 CONTINUE
      CLOSE(3)
      WRITE(*,*)'SMOOTHED K-PULSE SAVED IN FILE ',FNZ
C
C     QUERY TO DO ANOTHER RUN
C
   95 CONTINUE
      WRITE(*,*)
```

```
      WRITE(*,100) BELL
      WRITE(*,*) 'DO YOU WISH TO SMOOTH ANOTHER K-PULSE?'
      WRITE(*,*)
      READ(*,100) ANS
      IF((ANS.EQ.'Y').OR.(ANS.EQ.'y')) THEN
           GO TO 10
      ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n')) THEN
           GO TO 99
      ELSE
           WRITE(*,*) 'TRY AGAIN! "Y" OR "N".'
           GO TO 95
      ENDIF
 99   CONTINUE
      WRITE(*,100) BELL
      WRITE(*,*) 'CONVOLUTION PROGRAM IS OVER.  CHECK NEW FILE(S)'
100   FORMAT(A)
120   FORMAT(I5)
130   FORMAT(E12.6)
      STOP
      END
```

APPENDIX E. K-PULSE CONVOLUTION PROGRAM

A. PROGRAM DESCRIPTION

This program performs the time convolution of the normalized smoothed K-Pulse and a normalized input signal and computes the late-time energy of the output signal.

The program reads the time sequences of the K-Pulse and the input signal and stores them in arrays. A time convolution without zero padding is then performed. The late-time energy is calculated based on the value of the convolution output late-time start specified by the user. Each convolution output has its specific late-time start, based upon the length of the input signal early-time component and the length of the K-Pulse. The late-time energy is displayed on the computer screen, and the convolution time sequence is stored in a user specified output file.

B. PROGRAM LISTING

The K-Pulse convolution program listing is included below.

```
      PROGRAM CONVOL
C
C     NOTES:
C
C          1) PROGRAM CONVOLVES TWO SIGNALS AND COMPUTES LATE TIME
C             ENERGY
C          2) INPUTS: TIME SERIES FROM TWO DATA FILES
C          3) CONVOLUTION SIGNAL OUPUT TO DATA FILE
C          5) ORIGINAL WRITTEN 8/87 IN FORTRAN 77 BY LT L. CHEEKS
C          6) UPDATED MAY 89 BY LT M.S. SIMON & PROF M.A. MORGAN
C
C     DECLARATIONS
C
      REAL X(1024),Y(1024),Z(2048)
      CHARACTER*1 ANS, BELL
      CHARACTER*8 FNX,FNY,FNZ
      CHARACTER*64 TITX,TITY,TITZ
```

```
C
      BELL=CHAR(7)
C
C     DATA FILE INPUT
C
   10 CONTINUE
      WRITE(*,*) 'THIS PROGRAM CONVOLVES X(n) * Y(n) = Z(n)'
      WRITE(*,*) 'AND PLACES Z(n) IN USER SPECIFIED OUTPUT FILE'
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF X(n) INPUT FILE'
      WRITE(*,*)
      READ(*,100) FNX
      WRITE(*,*)
      WRITE(*,*) 'X(n) INPUT FILE IS ',FNX
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF Y(n) INPUT FILE'
      WRITE(*,*)
      READ(*,100) FNY
      WRITE(*,*)
      WRITE(*,*) 'Y(n) INPUT FILE IS ',FNY
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF Z(n) FILE'
      WRITE(*,*)
      READ(*,100) FNZ
      WRITE(*,*)
      WRITE(*,*) 'OUTPUT FILE IS ',FNZ
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER HEADER TITLE FOR Z(n) FILE'
      WRITE(*,*)
      READ(*,100) TITZ
      WRITE(*,*)
C
C     READING INPUT FILES
C
      OPEN(1,FILE=FNX)
      OPEN(2,FILE=FNY)
      OPEN(3,FILE=FNZ)
      READ(1,100) TITX
      READ(1,120) NX
      READ(1,130) TX1
      READ(1,130) TX2
      DO 20 I=1,NX
           READ(1,130) X(I)
   20 CONTINUE
      WRITE(*,*) 'X(t) FILE READ'
      READ(2,100) TITY
      READ(2,120) NY
      READ(2,130) TY1
      READ(2,130) TY2
      DO 30 I=1,NY
           READ(2,130) Y(I)
```

```
   30 CONTINUE
      WRITE(*,*) 'Y(t) FILE READ'
      WRITE(*,*)
      CLOSE(1)
      CLOSE(2)
C
C     CONVOLUTION ALGORITHM WITHOUT ZERO PADDING
C     Written by M.A. Morgan 5/19/89
C
      WRITE(*,*) 'CONVOLUTION IN PROGRESS'
      WRITE(*,*)
C
      DO 50 N=1,NY
      Z(N)=0.0
      M2=MIN0(N,NX)
      DO 40 M=1,M2
   40 Z(N)=Z(N)+X(M)*Y(N-M+1)
   50 CONTINUE
      N1=NY+1
      N2=NX+NY-1
      DO 70 N=N1,N2
      Z(N)=0.0
      M1=N+1-NY
      M2=MIN0(N,NX)
      DO 60 M=M1,M2
   60 Z(N)=Z(N)+X(M)*Y(N-M+1)
   70 CONTINUE
C
      WRITE(*,*) 'SAVING Z(t) TO OUTPUT DATA FILE'
      WRITE(3,100) TITZ
      WRITE(3,120) N2
      TZ1=TX1+TY1
      TZ2=TX2+TY2
      WRITE(3,130) TZ1
      WRITE(3,130) TZ2
      DO 80 I=1,N2
           WRITE(3,130) Z(I)
   80 CONTINUE
      CLOSE(3)
C
C     ROUTINE TO CALCULATE LATE-TIME ENERGY
C
   83 CONTINUE
      WRITE(*,*)
      WRITE(*,*) 'ENTER TL (Nsec) FOR LATE-TIME ENERGY CALCULATION'
      WRITE(*,*)
      READ(*,*) TLATE
      NLATE=NINT(TLATE*N2/(TZ2-TZ1))+1
      IF(NLATE.LT.N2) GO TO 85
      WRITE(*,100) BELL
      WRITE(*,*) 'TLATE TOO LARGE - TRY AGAIN'
      GO TO 83
   85 CONTINUE
      IF(NLATE.GE.1) GO TO 87
      WRITE(*,100) BELL
```

```
      WRITE(*,*) 'TLATE TOO SMALL - TRY AGAIN'
      GO TO 83
   87 CONTINUE
      ENERGY=0.0
      DO 90 N=NLATE,N2
   90 ENERGY=ENERGY+Z(N)*Z(N)
      ENERGY=ENERGY/(N2-NLATE+1)
      WRITE(*,*)
      WRITE(*,*) 'LATE TIME ENERGY =', ENERGY
C
C     QUERY TO DO ANOTHER RUN
C
   95 CONTINUE
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'DO YOU WISH TO CONVOLVE ADDITIONAL WAVEFORMS?'
      WRITE(*,*)
      READ(*,100) ANS
      IF((ANS.EQ.'Y').OR.(ANS.EQ.'y')) THEN
           GO TO 10
      ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n')) THEN
           GO TO 99
      ELSE
           WRITE(*,*) 'TRY AGAIN! "Y" OR "N".'
           GO TO 95
      ENDIF
   99 CONTINUE
      WRITE(*,100) BELL
      WRITE(*,*) 'CONVOLUTION PROGRAM IS OVER.  CHECK NEW FILE(S)'
  100 FORMAT(A)
  120 FORMAT(I5)
  130 FORMAT(E12.6)
      STOP
      END
```

APPENDIX F. E-PULSE GENERATOR PROGRAM

A. PROGRAM DESCRIPTION

This program generates E-Pulses using rectangular pulse basis functions that are used to annihilate input target return signals. The interactive program format of the program requires the following user specified inputs:

- the number of pole pairs for the E-Pulse;
- the scale factor (for scaled targets);
- the damping coefficient and the angular frequency of each signal pole;
- the width of the basis functions;
- the number of samples to be used for the forcing component;
- the number of samples to be used for the extinction component.

The program reads the inputs, scales the poles, forms the complex variable sn, and the matrices of Equation 2.14b. The basis function amplitudes are calculated by a matrix inversion subroutine and are then normalized by their RMS energy. The normalized amplitudes are stored in a user specified output file.

B. PROGRAM LISTING

The E-Pulse generator program listing is included below.

```
C       PROGRAM EP
C
C       PROGRAM NOTES
C       1)  PROGRAM GENERATES E-PULSES
C       2)  INPUTS:  POLES(REAL/IMAG)
C       3)  OUTPUT TO DATA FILE:  COMPLEX E-PULSE TIME SERIES
C       4)  PROGRAM REQUIRES MATH COPROCESSOR
C       5)  PROGRAM WRITTEN IN MSFORTRAN JUL89 BY LT M.S. SIMON
C           WITH PROF M.A. MORGAN
C       DECLARATIONS
        REAL T,TE,DT,SFAC,FMAX,CHECKT,RE(100),IM(100),RLEPTS(1024)
```

```
     +,IMEPTS(1024),SMPL,TZ/0.0/,LP
      INTEGER N,NN,NMX,M,MMX,H,I,NF,NE,Z,P(100),J,K,ZZ
      COMPLEX ESUM,E(100),SN(100),F(100,100),FP(100,100),DETERM
      COMPLEX D(100),B(100,100),Y(100),ALFA,E0,E2(100),GSUM
      CHARACTER FN*8,TITLE*64,ANS*1
C
C     ***MAIN BODY OF PROGRAM***
C
C     ***DATA INPUT ROUTINE***
C
 1    WRITE(*,*) 'E-PULSE GENERATOR'
      WRITE(*,*) 'ENTER # OF POLE PAIRS'
      READ(*,320) Z
      WRITE(*,*) '# OF POLE PAIRS = ',Z
      WRITE(*,*) 'ENTER SCALE FACTOR FOR TARGET SIZE/STANDARD TARGET'
      READ(*,*) SFAC
      WRITE(*,*) 'SCALE FACTOR = ',SFAC
25    DO 30 I=1,Z
           WRITE(*,*) 'ENTER REAL PART OF POLE #',I
           READ(*,*) RE(I)
           WRITE(*,*) 'REAL PART = ',RE(I)
           WRITE(*,*) 'ENTER IMAG PART OF POLE #',I
           READ(*,*) IM(I)
           WRITE(*,*) 'IMAG PART = ',IM(I)
30    CONTINUE
C
C   *****ROUTINE TO DETERMINE DT AND PULSE WIDTHS*****
C
      ZZ=2*Z
      FMAX=0.0
      DO 40 I=1,Z
         FMAX=AMAX1(FMAX,IM(I))
40    CONTINUE
      WRITE(*,*)'ENTER DELTA T IN NANOSECONDS'
      READ(*,*) DT
50    WRITE(*,*)'ENTER NUMBER OF SAMPLES TO BE USED TO CREATE'
      WRITE(*,*)'FORCING COMPONENT OF E-PULSE'
      WRITE(*,*)'SAMPLES ARE TAKEN 1 PER ',DT, ' NANOSEC'
      READ(*,*)  NF
      WRITE(*,*)  ' '
      TF=DT*NF
      WRITE(*,*)'FORCING COMPONENT IS ',TF,' NANOSEC IN LENGTH'
      WRITE(*,*)' '
      WRITE(*,*)'ENTER NUMBER OF SAMPLES TO BE USED TO CREATE EACH'
      WRITE(*,*)'SECTION OF THE EXTINCTION COMPONENT OF THE E-PULSE'
      WRITE(*,*)'SAMPLES ARE TAKEN 1 PER ',DT,' NANOSEC'
      WRITE(*,*)'THIS E-PULSE WILL HAVE ',ZZ,' SECTIONS'
      READ(*,*) NE
      LT=DT*NE
      NN=ZZ*NE+NF
      TE=DT*(NN-1)
60    WRITE(*,*) 'ENTER OUTPUT FILE NAME (MAX 8 CHAR)'
      READ(*,300) FN
      WRITE(*,*) 'OUTPUT FILE NAME IS ',FN
      WRITE(*,*) 'ENTER OUTPUT TITLE (MAX 64 CHAR)'
```

```
      READ(*,310) TITLE
      WRITE(*,*) 'OUTPUT FILE IS ',TITLE
C
C     ***INPUT DATA CHECK***
C
      WRITE(*,*) 'INPUT DATA CHECK'
      WRITE(*,*) 'OUTPUT FILE NAME IS      ',FN
      WRITE(*,*) 'OUTPUT TITLE IS          ',TITLE
      WRITE(*,*) '# OF POLE PAIRS         = ',Z
      WRITE(*,*) 'TOTAL NUMBER OF SAMPLES IS ',NN
      WRITE(*,*) 'TOTAL LENGTH OF E-PULSE IS ',TE,' NANOSECONDS'
      WRITE(*,*) 'SCALE FACTOR            = ',SFAC
      DO 90 I=1,Z
           WRITE(*,*) 'REAL PART OF POLE # ',I,' = ',RE(I)
           WRITE(*,*) 'IMAG PART OF POLE # ',I,' = ',IM(I)
           PAUSE
90    CONTINUE
91    WRITE(*,*) 'IS DATA CORRECT?'
      READ(*,290) ANS
      IF((ANS.EQ.'Y').OR.(ANS.EQ.'y'))THEN
           WRITE(*,*) 'DATA CORRECT'
           GO TO 92
      ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n'))THEN
           WRITE(*,*) 'DATA ERRORS. START AGAIN'
           GO TO 1
      ELSE
           WRITE(*,*) 'TRY AGAIN! Y OR N'
           GO TO 91
      ENDIF
C
C     ***CALCULATIONS***
C
92    WRITE(*,*) 'SCALING POLES AND FORMING COMPLEX VARIABLES'
C
C     ***ROUTINE TO SCALE REAL AND IMAG PARTS OF POLES***
C
      DO 93 I=1,Z
           RE(I)=RE(I)/SFAC
           IM(I)=IM(I)/SFAC
93    CONTINUE
C
C     ***ROUTINE TO FORM COMPLEX CONJUGATES OF INPUT POLES***
C
      DO 100 I=1,Z
           RE(I)=RE(I)
           RE(Z+I)=RE(I)
           IM(I)=IM(I)
           IM(Z+I)=-IM(I)
100   CONTINUE
C
C     ***ROUTINE TO CREATE COMPLEX VARIABLE SN***
C
      DO 110 J=1,ZZ
           SN(J)=CMPLX(RE(J),IM(J))
110   CONTINUE
```

```
C
C     ***ROUTINE TO FORM MATRIX F***
C
125   WRITE(*,*) 'ROUTINE TO FORM MATRIX F'
      DO 130 H=1,ZZ
           DO 128 I=1,ZZ+1
            F(H,I)=CEXP(-SN(H)*I*DT)
128        CONTINUE
130   CONTINUE
C
C     *****ROUTINE TO INVERT MATRIX F AND SOLVE FOR*****
C     ********AMPLITUDES OF E-PULSE EXTINCTION COMPONENTS********
C
      WRITE(*,*)'ROUTINE TO INVERT MATRIX F AND SOLVE FOR'
      WRITE(*,*)'AMPLITUDES OF E-PULSE EXTINCTION COMPONENTS'
      NMX=100
      MMX=100
      M=ZZ
      N=ZZ
      DO 145 H=1,ZZ
         B(H,1)=(-1.0,0.0)
145   CONTINUE
      CALL FACTOR(F,P,D,N,NMX)
      WRITE(*,*)'FACTOR COMPLETE'
      CALL INVERT(F,B,P,D,N,NMX,M,MMX)
      WRITE(*,*)'INVERT COMPLETE'
C
C    **ROUTINE TO FORM E-PULSE WITH BASIS FUNCTIONS**
C                 ** OF CORRECT AMPLITUDE**
C
      K=NF
      L=NF+NE-1
      DO 150 I=0,NN-1
         E(I)=(0.0,0.0)
150   CONTINUE
      DO 160 I=0,NF-1
      E(I)=(1.0,0.0)
160   CONTINUE
      DO 180 J=1,ZZ
         ALFA=B(J,1)
         DO 170 I=K,L
            E(I)=ALFA
170      CONTINUE
      K=K+NE
      L=L+NE
180   CONTINUE
C
C     ***ROUTINE TO NORMALIZE E-PULSE TIME SERIES***
C
      GSUM=(0.0,0.0)
      DO 190 I=0,NN-1
         E2(I)=E(I)**2
         GSUM=GSUM+E2(I)
190   CONTINUE
      E0=SQRT(GSUM)
```

```
      DO 195 I=0,NN-1
         E(I)=E(I)/E0
195   CONTINUE
      DO 200 I=0,NN-1
        RLEPTS(I)=REAL(E(I))
        IMEPTS(I)=AIMAG(E(I))
200   CONTINUE
      WRITE(*,*) 'CALCULATION ROUTINE IS COMPLETE'
C
C     ***ROUTINE TO STORE E-PULSE TIME SERIES IN DATA FILE***
C
      WRITE(*,*) 'STORING E-PULSE IN DATA FILE'
      OPEN(1,FILE=FN)
      WRITE(1,310) TITLE
      WRITE(1,320) NN
      WRITE(1,330) TZ
      WRITE(1,330) TE
      DO 240 I=0,NN-1
           WRITE(1,340) RLEPTS(I),IMEPTS(I)
240   CONTINUE
      CLOSE(1)
      WRITE(*,*) 'E-PULSE TIME SERIES STORED IN FILE ',FN
C
C     ***ROUTINE TO DO ANOTHER RUN***
C
      WRITE(*,*) 'DO YOU WISH TO GENERATE ANOTHER E-PULSE?'
245   READ(*,290) ANS
      IF((ANS.EQ.'Y').OR.(ANS.EQ.'y'))THEN
           WRITE(*,*) 'OK. ANOTHER E-PULSE'
           GO TO 1
           ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n'))THEN
           WRITE(*,*) 'OK'
           GO TO 280
      ELSE
           WRITE(*,*) 'TRY AGAIN! Y OR N'
           GO TO 245
      ENDIF
280   WRITE(*,*) 'E=PULSE GENERATOR PROGRAM COMPLETE'
      WRITE(*,*) 'CHECK NEW FILE(S)'
290   FORMAT(A1)
300   FORMAT(A8)
310   FORMAT(A64)
320   FORMAT(I5)
330   FORMAT(E12.6)
340   FORMAT(E12.6,3X,E12.6)
400   STOP
      END
C
C     *********SUBROUTINE FACTOR***********
C     *****WRITTEN BY PROF M.A. MORGAN*****
C
       SUBROUTINE FACTOR (A,P,D,N,NMX)
       DIMENSION A(NMX,NMX),D(NMX),P(NMX)
       COMPLEX A,D,DETER
       INTEGER R,P,RM1,RP1,PJ,PR
```

```
      DO 60 R=1,N
      DO 10 K=1,N
      D(K)=A(K,R)
 10   CONTINUE
      RM1=R-1
      IF(RM1.LT.1) GO TO 31
      DO 30 J=1,RM1
      PJ=P(J)
      A(J,R)=D(PJ)
      D(PJ)=D(J)
      JP1=J+1
      DO 20 I=JP1,N
      D(I)=D(I)-A(I,J)*A(J,R)
 20   CONTINUE
 30   CONTINUE
 31   CONTINUE
      DMAX=CABS(D(R))
      P(R)=R
      RP1=R+1
      IF(RP1.GT.N) GO TO 41
      DO 40 I=RP1,N
      ELMAG=CABS(D(I))
      IF(ELMAG.LT.DMAX) GO TO 40
      DMAX=ELMAG
      P(R)=I
 40   CONTINUE
 41   CONTINUE
      IF(DMAX.LT.1.0E-15) PRINT 105,DMAX,R
      PR=P(R)
      A(R,R)=D(PR)
      D(PR)=D(R)
      IF(RP1.GT.N) GO TO 51
      DO 50 I=RP1,N
      A(I,R)=D(I)/A(R,R)
 50   CONTINUE
 51   CONTINUE
 60   CONTINUE
      FNORM=0.0
      DETER=(1.,0.)
      DO 70 R=1,N
      DETER=DETER*A(R,R)
      DMAG=CABS(DETER)
      IF(DMAG.LT.1.0E10) GO TO 1
      DETER=DETER*1.0E-10
      FNORM=FNORM+10.
   1  CONTINUE
      IF(DMAG.GT.1.0E-10) GO TO 2
      DETER=DETER*1.0E10
      FNORM=FNORM-10.
   2  CONTINUE
      IF(ABS(FNORM).GT.9.0) PRINT 104,DMAG,FNORM,R
 70   CONTINUE
104   FORMAT (1H0,'CABS(DETER)= ',1PE12.3,' X 10 TO THE POWER ',
     1 0PF5.0,'  AT COLUMN  ',I3)
105   FORMAT (1H0,'MAXIMUM PIVOT = ',E13.2,' AT COLUMN ',I3)
```

```
      RETURN
      END
C
C     **********SUBROUTINE INVERT**********
C     *****WRITTEN BY PROF M.A. MORGAN*****
C
      SUBROUTINE INVERT (A,B,P,Y,N,NMX,M,MMX)
      DIMENSION A(NMX,NMX),B(NMX,MMX),P(NMX),Y(NMX)
      COMPLEX A,B,Y,SUM
      INTEGER P,PI
      DO 50 L=1,M
      DO 20 I=1,N
      PI=P(I)
      Y(I)=B(PI,L)
      B(PI,L)=B(I,L)
      IP1=I+1
      IF(IP1.GT.N) GO TO 11
      DO 10 J=IP1,N
      B(J,L)=B(J,L)-A(J,I)*Y(I)
   10 CONTINUE
   11 CONTINUE
   20 CONTINUE
      DO 40 K=1,N
      I=N-K+1
      SUM=(0.,0.)
      IP1=I+1
      IF(IP1.GT.N) GO TO 31
      DO 30 J=IP1,N
      SUM=SUM+A(I,J)*B(J,L)
   30 CONTINUE
   31 CONTINUE
      B(I,L)=(Y(I)-SUM)/A(I,I)
   40 CONTINUE
   50 CONTINUE
      RETURN
      END
```

APPENDIX G. E-PULSE CONVOLUTION PROGRAM

A. PROGRAM DESCRIPTION

This program performs the time convolution of the E-Pulse and a normalized input signal and computes the late-time energy of the output signal.

The program reads the time sequences of the E-Pulse and the input signal and stores them in arrays. The user specifies the integer multiple of the input signal sampling period that was used to form the rectangular pulse basis functions of the E-Pulse and a time convolution without zero padding is then performed. The late-time energy is calculated based on the value of the convolution output late-time start specified by the user. Each convolution output has its specific late-time start, based upon the length of the input signal early-time component and the length of the E-Pulse. The late-time energy is displayed on the computer screen, and the convolution time sequence is stored in a user specified output file.

B. PROGRAM LISTING

The E-Pulse convolution program listing is included below.

```
      PROGRAM CONE
C
C     NOTES:
C
C          1) PROGRAM CONVOLVES INPUT SIGNAL AND E-PULSE AND
C             COMPUTES LATE-TIME ENERGY
C          2) INPUTS: TIME SERIES FROM TWO DATA FILES
C          3) CONVOLUTION SIGNAL OUPUT TO DATA FILE
C          5) ORIGINAL WRITTEN 8/87 IN FORTRAN 77 BY LT L. CHEEKS
C          6) UPDATED MAY 89 BY LT M.S. SIMON & PROF M.A. MORGAN
C          7) HANDLES SIGNALS OF INTEGER MULTIPLES OF DT
C
```

```
C     DECLARATIONS
C
      REAL X(2048),Y(2048),Z(4096),SUMX(2048)
      CHARACTER*1 ANS, BELL
      CHARACTER*8 FNX,FNY,FNZ
      CHARACTER*64 TITX,TITY,TITZ
C
      BELL=CHAR(7)
C
C     DATA FILE INPUT
C
   10 CONTINUE
      WRITE(*,*) 'THIS PROGRAM CONVOLVES X(n) * Y(n) = Z(n)'
      WRITE(*,*) 'AND PLACES Z(n) IN USER SPECIFIED OUTPUT FILE'
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF X(n) INPUT FILE. THIS FILE HAS THE'
      WRITE(*,*) 'THE REFERENCE DT'
      WRITE(*,*)
      READ(*,100) FNX
      WRITE(*,*)
      WRITE(*,*) 'X(n) INPUT FILE IS ',FNX
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF Y(n) INPUT FILE(THE E PULSE)'
      WRITE(*,*)
      READ(*,100) FNY
      WRITE(*,*)
      WRITE(*,*) 'Y(n) INPUT FILE IS ',FNY
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER NAME OF Z(n) FILE'
      WRITE(*,*)
      READ(*,100) FNZ
      WRITE(*,*)
      WRITE(*,*) 'OUTPUT FILE IS ',FNZ
      WRITE(*,*)
      WRITE(*,100) BELL
      WRITE(*,*) 'ENTER HEADER TITLE FOR Z(n) FILE'
      WRITE(*,*)
      READ(*,100) TITZ
      WRITE(*,*)
C
C     READING INPUT FILES
C
      OPEN(1,FILE=FNX)
      OPEN(2,FILE=FNY)
      OPEN(3,FILE=FNZ)
      READ(1,100) TITX
      READ(1,120) NX
      READ(1,130) TX1
      READ(1,130) TX2
      DO 20 I=1,NX
           READ(1,130) X(I)
   20 CONTINUE
```

```
      WRITE(*,*) 'X(t) FILE READ'
      READ(2,100) TITY
      READ(2,120) NY
      READ(2,130) TY1
      READ(2,130) TY2
      DO 30 I=1,NY
           READ(2,130) Y(I)
   30 CONTINUE
      WRITE(*,*) 'Y(t) FILE READ'
      WRITE(*,*)
      CLOSE(1)
      CLOSE(2)
      WRITE(*,*) 'ENTER THE VALUE OF "K" THE MULTIPLE OF DT'
      READ(*,*) K
C
C     CONVOLUTION ALGORITHM WITHOUT ZERO PADDING
C     Written by LT M.S. Simon 8/4/89
C
      WRITE(*,*) 'CONVOLUTION IN PROGRESS'
      WRITE(*,*)
C
      DO 50 N=1,K*NY
         Z(N)=0.0
         R=MOD(N,K)
         IF(R.EQ.0.0) THEN
            L=N/K
         ELSE
            L=INT(N/K)+1
         ENDIF
         M2=MIN0(L,NX)
         DO 40 M=1,M2
            SUMX(M)=0.0
            J1=N-(M*K)+1
            J2=N-((M-1)*K)
            DO 35 J=J1,J2
               SUMX(M)=SUMX(M)+X(J)
               IF(J.EQ.0.0) THEN
                  SUMX(M)=0.0
               ELSE
                  SUMX(M)=SUMX(M)
               ENDIF
35          CONTINUE
            Z(N)=Z(N)+SUMX(M)*Y(M)
40       CONTINUE
50    CONTINUE
      N1=K*NY+1
      N2=NX+K*NY-1
      DO 70 N=N1,N2
         Z(N)=0.0
         M1=((N+1)-(K*NY))
         DO 60 M=1,NY
            SUMX(M)=0.0
            J3=M1+((M-1)*K)
            J4=M1+((M*K)-1)
            DO 55 J=J3,J4
```

```
                 SUMX(M)=SUMX(M)+X(J)
55            CONTINUE
              Z(N)=Z(N)+SUMX(M)*Y(NY-M+1)
60         CONTINUE
70      CONTINUE
C
        WRITE(*,*) 'SAVING Z(t) TO OUTPUT DATA FILE'
        WRITE(3,100) TITZ
        WRITE(3,120) N2
        TZ1=TX1+TY1
        TZ2=TX2+TY2
        WRITE(3,130) TZ1
        WRITE(3,130) TZ2
        DO 80 I=1,N2
             WRITE(3,130) Z(I)
  80    CONTINUE
        CLOSE(3)
C
C       ROUTINE TO CALCULATE LATE-TIME ENERGY
C
  83    CONTINUE
        WRITE(*,*)
        WRITE(*,*) 'ENTER TL (Nsec) FOR LATE-TIME ENERGY CALCULATION'
        WRITE(*,*)
        READ(*,*) TLATE
        NLATE=NINT(TLATE*N2/(TZ2-TZ1))+1
        IF(NLATE.LT.N2) GO TO 85
        WRITE(*,100) BELL
        WRITE(*,*) 'TLATE TOO LARGE - TRY AGAIN'
        GO TO 83
  85    CONTINUE
        IF(NLATE.GE.1) GO TO 87
        WRITE(*,100) BELL
        WRITE(*,*) 'TLATE TOO SMALL - TRY AGAIN'
        GO TO 83
  87    CONTINUE
        ENERGY=0.0
        DO 90 N=NLATE,N2
  90    ENERGY=ENERGY+Z(N)*Z(N)
        ENERGY=ENERGY/(N2-NLATE+1)
        WRITE(*,*)
        WRITE(*,*) 'LATE TIME ENERGY =', ENERGY
C
C       QUERY TO DO ANOTHER RUN
C
  95    CONTINUE
        WRITE(*,*)
        WRITE(*,100) BELL
        WRITE(*,*) 'DO YOU WISH TO CONVOLVE ADDITIONAL WAVEFORMS?'
        WRITE(*,*)
        READ(*,100) ANS
        IF((ANS.EQ.'Y').OR.(ANS.EQ.'y')) THEN
             GO TO 10
        ELSEIF((ANS.EQ.'N').OR.(ANS.EQ.'n')) THEN
             GO TO 99
```

```
      ELSE
           WRITE(*,*) 'TRY AGAIN! "Y" OR "N".'
           GO TO 95
      ENDIF
 99   CONTINUE
      WRITE(*,100) BELL
      WRITE(*,*) 'CONVOLUTION PROGRAM IS OVER.  CHECK NEW FILE(S)'
 100  FORMAT(A)
 120  FORMAT(I5)
 130  FORMAT(E12.6)
      STOP
      END
```

LIST OF REFERENCES

1. Jean, Y. K., *Target Identification Using Natural Resonance and All-Zero Filter Cancellation* Techniques, Master's Thesis, Naval Postgraduate School, Monterey, California, April 1984.

2. Dunavin, J. B., *Identification of Scatterers Based Upon Annihilation of Complex Natural Resonances*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1985.

3. Mains, R. H., and D. L. Moffatt, *Complex Natural Resonances of an Object in Detection and Discrimination*, Ohio State University, Electroscience Laboratory, Columbus, Ohio, June 1974.

4. Baum, C. E., "The Singularity Expansion Method," *Transient Electromagnetic Fields*, L.B. Felson, Ed., Springer-Verlag, New York, 1976, pp.130-176.

5. Morgan, M. A., "Scatterer Discrimination Based Upon Natural Resonance Annihilation," *Journal of Electromagnetic Waves and Applications*, v. 2, no. 5/6, 1988, pp. 481-502.

6. Choong, C. Y., *Investigation of Non-Linear Estimation of Natural Resonances in Target Identification*, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1983.

7. Manhila, F. M. C., *Investigation of Methods of Natural Resonance Radar Target Identification*, Master's Thesis, Naval Postgraduate School, Monterey, California, October 1982.

8. Davenport, R. W., *Target Identification Using Prony's Method*, Master's Thesis, Naval Postgraduate School, Monterey, California, October 1982.

9. Kennauagh, E. M., "K-Pulse Concept," *IEEE Transactions on Antennas and Propagation*, v. AP-29, no. 3, March 1981, pp. 327-332.

10. Morgan, M.A., and Dunavin, J. B. "Discrimination of Scatterers Using Natural Resonance Annihilation," *Abstracts of 1986 National Radio Science Meeting*, Philadelphia, Pennsylvania, June 1986.

11. Chen, K.M. and others, "Radar Target Discrimination by Convolution of Radar Return with Extinction-Pulses and Single-Mode Extraction Signals," *IEEE Transactions on Antennas and Propagation*, v. AP-34, no. 7, July 1986, pp.896-904.

12. Chen, K.M. and others, *Aspect-Independent Target Discrimination Using Discriminant Signals*, Department of Electrical Engineering and Systems Science, Michigan State University, East Lansing, Michigan.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	1
4.	Professor Joseph Sternberg, Code 56Sn Electronic Warfare Academic Group Naval Postgraduate School Monterey, CA 93943-5100	1
5.	Professor M.A. Morgan, Code 62Mw Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	10
6.	CDR J.R. Powell, USN, Code 62Pl Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	1
7.	Professor R.W. Adler, Code 62Ab Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	1

8. Dr. John N. Entzminger 1
Director, Tactical Technology Office
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

9. Dr. Dominick Giglio 1
Tactical Technology Office
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

10. Dr. Rabiner Madan 1
Code 1114SE
Office of Naval Research
800 N. Quincy St.
Arlington, VA 22209

11. Professor K.M. Chen 1
Department of Electrical Engineering
and Systems Science
Michigan State University
East Lansing, MI 48824

12. Dr. M.L. Van Blaricum 1
Toyon Research Corporation
75 Aero Camino, Suite A
Goleta, CA 93117

13. Mr. Daniel Carpenter 1
TRW Military Electronics and Aviation Division
One Rancho Carmel
San Diego, CA 92128

14. Dr. C. Ray Smith 1
AMSMI-RD-AS-RA
U.S. Army Missile Command
Redstone Arsenal, AL 35898-5253

15. LT M.S. Simon, USN 1
Surface Warfare Officer School Command
Department Head Course Class 111
Naval Education and Training Center
Newport, RI 02843-5000